水理・水文解析のための汎用プラットフォームの開発に関する研究

菊　森　佳　幹

目　次

第 1 章　序論

1-1 研究の経緯

我が国では，計算機の発達や数値解析技術の進歩とともに，数多くの水理・水文解析ソフトウェアが開発されてきた．これらは通常，研究機関ごと，企業あるいは研究者ごとに独自に知見や工夫を導入し，水理・水文現象の解明等の研究および河川事業等の実務に用いられてきている．

しかし，これらの国内の水理・水文解析ソフトウェアは，それぞれが独自の入出力インターフェースをもつため，他の解析ソフトウェアと連携利用ができない，インターネット上で公開されているデータベース上の河川流域に関する水理・水文データ等を容易には活用できない，開発者以外がソフトウェアを使うのは困難である等，連携性や操作性の課題を有している（図-1.1）．

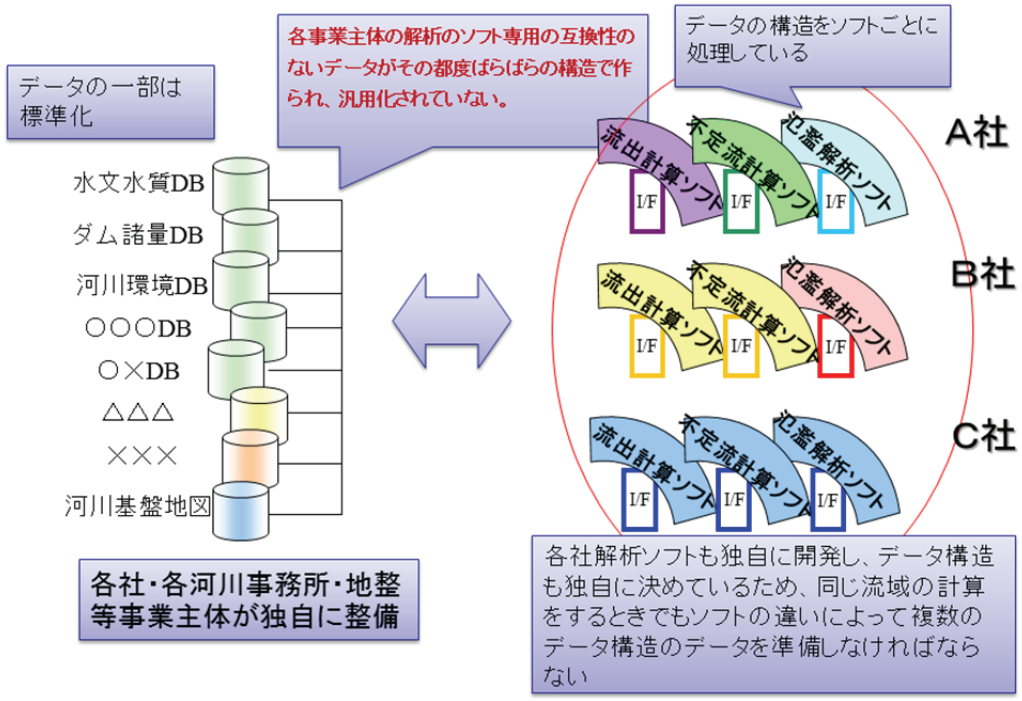


図-1.1 水理・水文解析ソフトウェアや河川・流域に関するデータを取り巻く状況

また，国や地方公共団体の職員数減少に伴い，現場の河川技術者が調査に費やすことができる時間が少なくなり，他者が簡単に使える水理・水文解析ソフトウェアがないこともあり，自ら解析することがほとんどなくなった．その結果，技術力の低下や水理・水文デ

ータの品質管理への動機づけが低下する方向に向かっている．国や地方公共団体では，水理・水文解析はコンサルタントに業務を外注することが主流になっているが，コンサルタントの用いる解析ソフトウェアの計算方法が明らかになっていなかったりすることや自らが解析ソフトウェアを持っていないため,再現計算して解析結果を確認することができず,場合によっては治水計画等の根拠を説明するのが困難な状況になることもある．研究分野においても，解析プログラムの他者利用が事実上できないため，物質輸送を解析するプログラムを水量の解析プログラムと組み合わせる，あるいは河道流下と氾濫の解析プログラムを組み合わせて用いるといった利用に非常に手間がかかるため，この分野の研究が十分に発展しない弊害が見られる（図-1.2）．

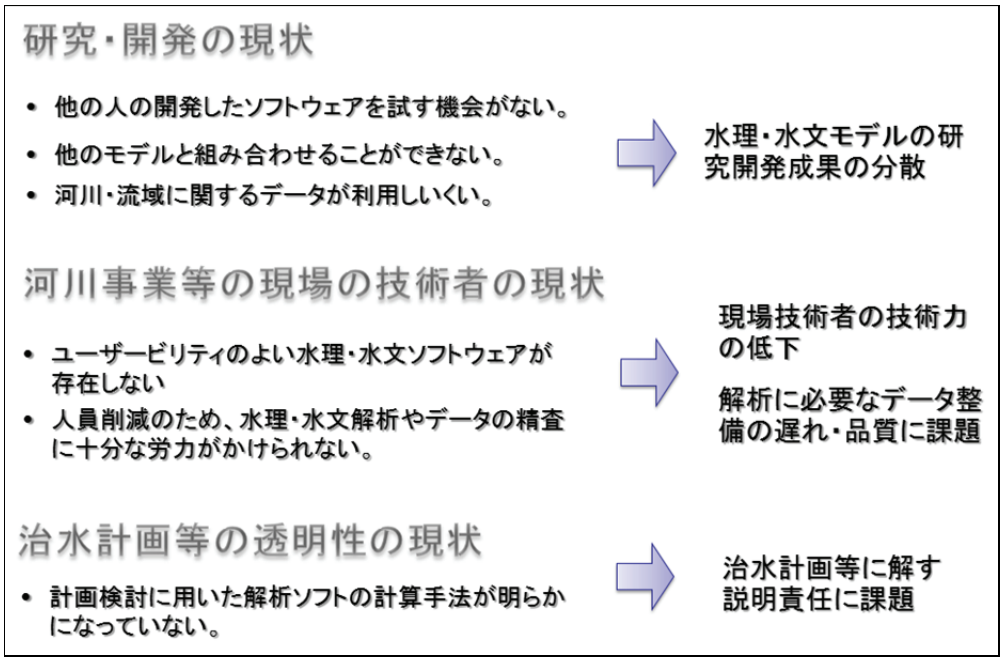
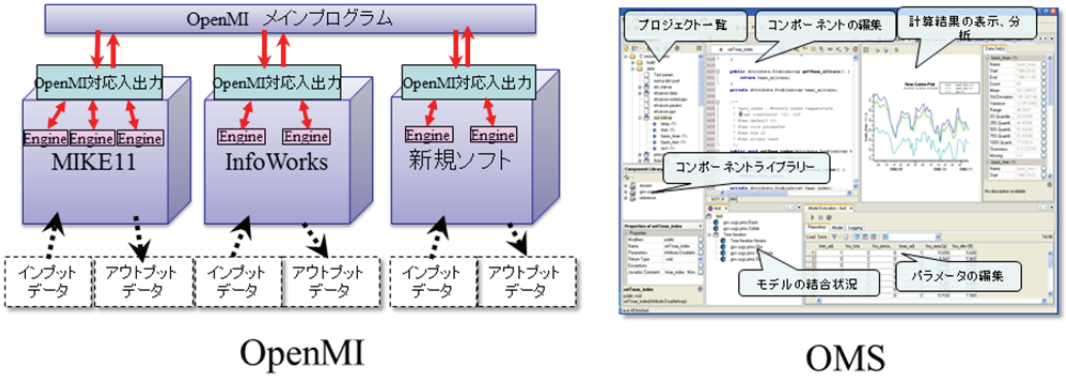


図-1.2 水理・水文解析に関する研究・開発や河川技術者等に関する状況

一方，欧州ではデンマーク水理環境研究所等の世界的な主流となっている水理・水文解析ソフトウェアを開発している会社や EC（欧州共同体）等の出資により開発した OpenMI があり，また米国では米農政務省が中心となって開発している OMS（Object Modeling System）があり，欧米ではすでに異なる開発者の複数の水理・水文解析モデルを連接させるための基盤となるソフトウェア（以下,「汎用プラットフォーム」と呼ぶ）が利用されている状況である（図-1.3）．国内の個々の水理・水文解析ソフトウェアはその解析性能において，必ずしも劣っているわけではないが，基盤となるための広く使われている汎用プラ

ットフォームがなかったので，国内の知見を集結させることができず，技術援助では欧米ソフトウェアに席巻され，海外での実績は乏しいものであった．特に最近では OpenMI のインターフェース仕様を OGC（Open Geospatial Consortium）の国際標準にしようとする動きがあり，戦略的な対応の必要性が高まってきている．



欧州では、水理・水文解析メーカや欧州共同体の出資により、米国では農政務省等により、水理・水文解析のための汎用プラットフォームが開発されている。

図-1.3 欧米の汎用プラットフォーム

このような認識は古くから多くの人々の間で醸成されてきていた[1]．汎用的な水文モデルの構築システムである京都大学の OHyMoS[2]の開発が 1990 年代初頭から行われており，また，河川技術者自らが解析を行うための水理水文解析ソフトウェアである（財）国土技術研究センターの「河道計画シミュレーター」[3]の開発等の取り組みがあった．国土技術政策総合研究所（国総研）においては，2002 年度から 2006 年度まで，「水理・水文・水質シミュレーションモデルの開発戦略に関する調査研究」が行われた[1]．その成果を受け，2007 年度から国総研および国土交通省河川局（当時）や土木学会が連携して水理・水文モデルの汎用プラットフォーム（水・物質循環解析ソフトウェア共通プラットフォーム，以下「CommonMP」と呼ぶ）の開発と利用を促進するプロジェクト[4]が 2007 年 4 月に開始された．国総研は同 9 月に CommonMP ウェブサイト[5]を開設し，継続して本プロジェクトの取り組みの情報発信を行っている（図-1.4）．2009 年 7 月には，国土交通省河川局（当時），同都市・地域整備局下水道部（当時），国総研，土木学会，建設コンサルタンツ協会および全国上下水道コンサルタント協会の産官学連携のコンソーシアム（CommonMP 開発・運営コンソーシアム）[6]を結成し，本プロジェクトの一部を担うようになった．

図-1.4 水・物質循環解析ソフトウェア共通プラットフォームのウェブサイト [5)]

本研究は，構想段階から CommonMP 初版公表までの間に，本プロジェクトの取組の一部として行った国内における水理水文解析ソフトウェアや河川技術者等に関わる課題の分析，国内外の既存の汎用プラットフォームの開発の経緯や開発仕様の調査，それらの汎用プラットフォームの国内の課題に対する適応性の分析，国内の課題を解決するための CommonMP の開発方針と機能要件の分析，および機能要件を実現するためのプログラム設計・実装を行うものである。

1-2 論文の構成

本論文は，以下の 6 章から構成される．

第 1 章序論は，水理・水文解析ソフトウェアの開発に関する既往の検討経緯をまとめたものである．

第 2 章は「水理・水文解析を取り巻く状況」と題して，水理・水文解析に用いる水文データや河道断面データ等の整備状況，国内の既存の水理・水文解析ソフトウェアの開発状況，水理・水文解析に関する人材や組織の状況について調査するとともに，既存の水理・水文解析のための汎用プラットフォームの開発状況に関する調査に基づき，国内における

水理・水文解析を取り巻く課題を分析したものである．

第 3 章は「既存の汎用プラットフォームの開発状況」と題して，国内外の汎用プラットフォームの開発経緯や仕様を調査するとともに，第 2 章で分析した国内における水理・水文解析を取り巻く課題への適応性について分析したものである．

第 4 章は「汎用プラットフォームの開発目的および機能要件の分析」と題して，第 3 章の既存の汎用プラットフォームの分析結果を踏まえて，新たに開発する汎用プラットフォームの開発目的と開発方針を定めた上，第 2 章で分析した国内における水理・水文解析を取り巻く課題に対応するための汎用プラットフォームの開発目的および機能要件を分析したものである．

第 5 章は「プログラムの設計・実装」と題し，第 4 章で分析した機能要件を実現できるように汎用プラットフォームのプログラムを設計したものである．想定されるユーザや要素モデルの開発者の属性を考慮して開発プログラミング言語，開発環境，実行環境等を設定した．次に機能要件を満たすようなシミュレーション・プロジェクトの演算制御方式や要素モデルの演算実施方法等の設計を実施した．また，Fortran プログラムのためのラッピング手法や複数シミュレーション・プロジェクトの同時並行稼動方式，プロジェクト間フィードバックの方法を設計・実装し，性能試験を行った．

第 6 章結論では，主要な結論と今後の課題を論述している．

第 2 章　水理・水文解析を取り巻く状況 [7)]

2-1 概説

本章では，水理・水文解析に必要となる河川・流域に関する水文データや地形データの整備状況，シミュレーションを実施する水理・水文解析ソフトウェアの開発状況，水理・水文解析を実施する，あるいは利用する河川技術者等の人材やその所属組織について調査し，日本国内における水理・水文解析を取り巻く課題を分析する．

2-2 水文・水文解析に用いるデータの状況

水理・水文シミュレーションを実施するためには，降雨や水位等の水文データ，および河道断面等の地形データが必要となる．河川･流域内のこれらのデータは種々存在するが，ここでは，河川事業で主に用いられる水文データと河道断面データの整備状況について調査することとする．

水文データについては，国土交通省所管の観測所の水文観測データ（雨量，水位，流量，水質等）が水文水質データベース [8)]に格納され，インターネットを経由してウェブブラウザで閲覧およびダウンロードできるようになっている．2012 年 6 月現在，7,767 観測所のデータ（そのうち雨量観測所が 2,560 箇所，水位流量観測所が 2,065 箇所）が格納されている．テレメータ観測所データは，10 分ごとの観測値がリアルタイムで取得でき，過去 7 日間さかのぼって閲覧およびダウンロードすることができる．一方，毎正時のデータは永久保存される．観測後保存された毎正時データは暫定値とされ，水文観測業務規定 [9)]に基づき照査された後，確定値に差し替えられる．照査作業は，当該観測所を所管する地方整備局等において実施されている．

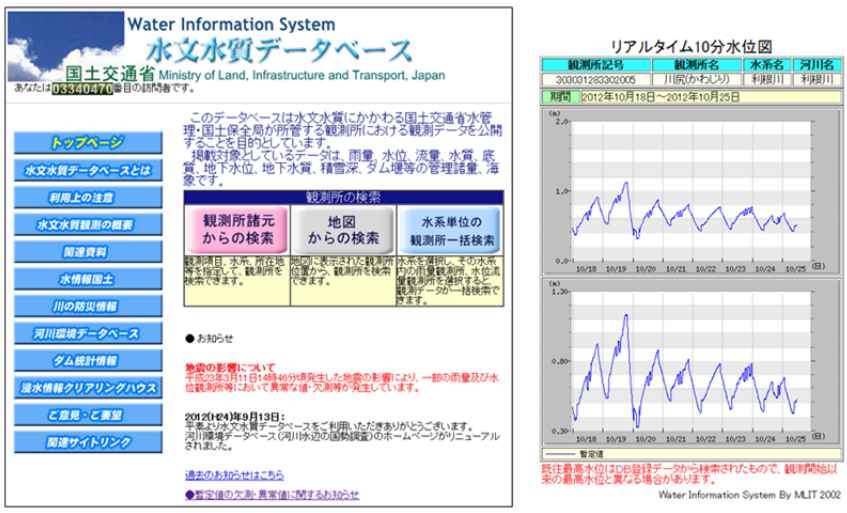


図-2.1　水文水質データベース [8)]

一方，「国土交通省リアルタイム川の防災情報」[10]には，国土交通省所管以外のテレメータ観測所データやレーダ雨量がリアルタイムで表示されている．雨量および水位のテレメータ観測所データは，過去３日間に渡り閲覧することができる．こちらは水文水質データベースのようにデータベースにデータを格納・保存しているわけではないので，それ以上の過去データは，参照することはできない．また，数値データをダウンロードすることができない仕様となっている．

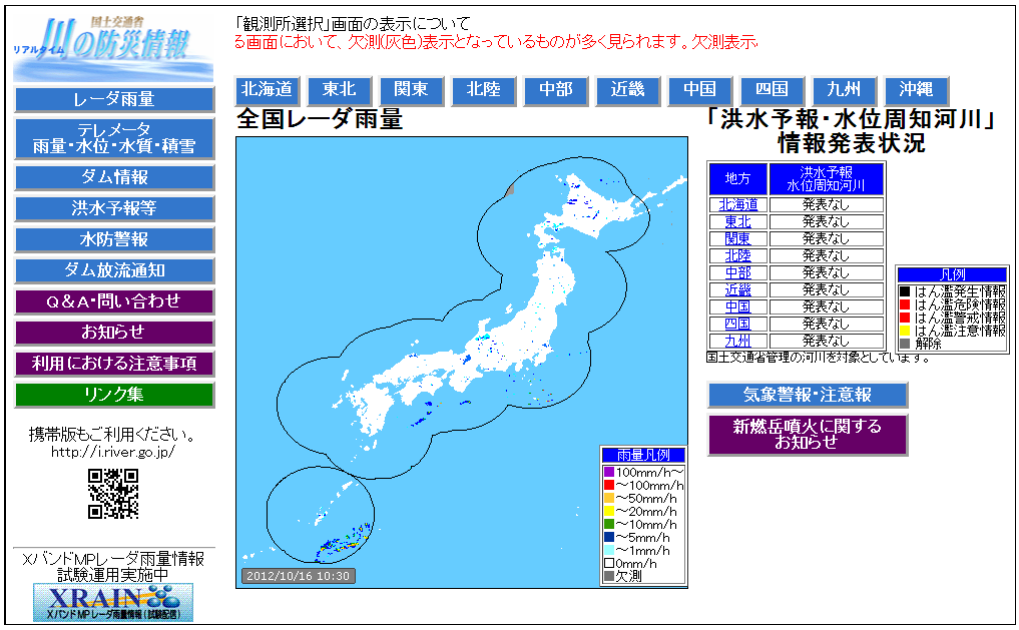


図-2.2　国土交通省【川の防災情報】[10]

国土交通省が所管する一級河川の管理区間の河道断面については，おおよそ５年に１度程度，河川定期縦横断測量が実施されている．2008 年より一級河川の直轄管理区間の定期縦横断測量データが水情報国土データ管理センター[11]の管理するデータベースに収集されつつある[12]．これらについては，それぞれの河川を所管する地方整備局の事務所により統一フォーマットにより提出された後，水情報国土データ管理センターにより照査され，河川縦横断データベースに登録されることとなっている．河川縦横断データベースは現在（2012 年６月）公開に向け，準備中である．

水文水質データベースおよび河川縦横断データベースには国土交通省が仕様を定めた標準インターフェース[13]を実装済みであり，標準インターフェースに対応したアプリケーション・ソフトウェアから直接ダウンロードすることができる．

ただし，河川定期縦横断測量データは，河道の急拡や急縮を水理解析に反映させる測定

間隔とはならない場合や，流心線に対して断面が垂直でない場合があり，水理解析に当たってはデータの補間や修正が必要な場合がある（図-2.3）．また，このデータは，河道の合流や分派等の接続状況に関する情報を含んでいないので，これのみでは河道ネットワーク・モデルを構築することはできない．

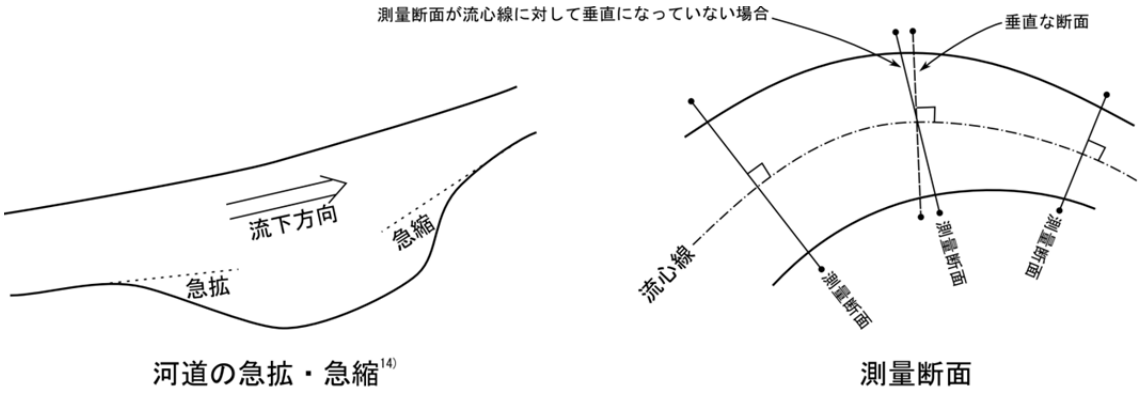


図-2.3 データの補間が必要な河道断面データ（河道の急拡・急縮等）

2-3　水理・水文解析に用いるソフトウェアの状況

本プロジェクトの一環として,筆者らは 2008 年に土木学会と連携して日本国内における水理・水文解析ソフトウェアの開発状況に関する調査を実施した．一部の海外のソフトウェアについても日本国内の代理店を通じて回答を得た．調査対象は，大学，河川コンサルタントおよび国土交通省出先機関であり，58 件の水理・水文解析ソフトウェアの回答を 33 の組織から得ている（表-2.1）．

表-2.1 得られた回答の内訳

	ソフトウェアの件数	組織数
民間組織	27	14
国・公益法人等	15	10
大学関係	16	9
合 計	58	33

回答から得られたソフトウェアのうち，複数の解析エンジン（モデル）を選択して，演算を行うことができるプラットフォームとしての機能があるものは 4 件あり，そのうち 3 件は海外の単独または連携プロジェクトであった．仕様を公開して不特定多数のモデル開

発者の参加を許容している汎用プラットフォームとしての機能を有しているのは，1 件のみ（OHyMoS[2)]）であった．

ソフトウェアの運用形態のタイプは，スタンドアロンで用いられるものと，リアルタイム洪水予測等のクライアント・サーバ方式でサーバ運用しているものに大別された．スタンドアロンで用いられるものは，GUI（Graphical User Interface）を持つものと，GUI を持たずにコマンドラインから操作するものに分類された．サーバ運用されるものは，特定の河川等の水位予測等に対応したものと，汎用的なものの２つのタイプに分類され，前者が 7 件であるのに対して，後者が 3 件であった．スタンドアロンで用いられるソフトウェアは，GUI を持つものと持たないものがあるが，前者は 20 件に対して，後者は 23 件であった（表-2.2）．GUI を持たないものについては，ユーザマニュアルの整備についての回答がなかった．

表-2.2 ソフトウェアの形態

	スタンドアロン型		サーバ運用		合 計
	GUI 付き	GUI なし	特定河川	汎用タイプ	
ソフトウェア件数	20	23	7	3	58

ソフトウェアの開発言語については，Fortran が 44 件，ついで C++が 20 件，Visual Basic が 12 件，C#が 5 件，Pascal が 3 件であった（表-2.3）．Visual Basic を用いたソフトウェアは，他の言語とともに用いていることが多いこと，およびすべて GUI を有するソフトウェアであることから，演算部分については主に Fortran や C++を用い，GUI 用に Visual Basic を用いていると推察できる．

表-2.3　開発言語の種類（件数・複数回答）

ソフトウェア / 言語の種類	スタンドアロン		サーバ運用	合計
	GUI付き	GUIなし		
Fortran	14	20	10	44
C++	10	9	1	20
Visual Basic	7	0	5	12
C#	3	2	0	5
Pascal	0	3	0	3

回答のうち，ソースコードを公開しているものは 8 件であり，大半はソースコードを公開していなかった．公開しているもののうち，7 件は大学に所属する者が開発したものであった．

ソフトウェアの著作権者は大学が 16 件，民間組織が 27 件，国や財団等公益法人が 15 件であった．著作権者が公益法人等であるものについては，民間組織が委託により開発したことが明記されているものがあった．民間組織が開発したソフトウェアで，大学等の協力を得たとの記述があったものは，3 件であった．

表-2.4　ソースコードの公開・非公開の別（件数）

公開・非公開 / 組織の種別	公開	非公開※	合計
大学	7	9	16
民間組織	1	26	27
国・公益法人等	0	15	15
合　計	8	50	58

※無回答を含む

本調査結果より，日本国内において不特定多数の解析モデルの開発者の参加を許容する汎用プラットフォームとしての機能をもつソフトウェアは，OHyMoS の他には見つからなかった．スタンドアロンで用いられるソフトウェアのうち，約 6 割が GUI を有しないソフトウェアであり，その大半がユーザマニュアルの整備に関して回答がなかった．これらは，特定のユーザのみで用いられているものと推測される．

また,大半のソフトウェアがソースコードの開示を明記してないことが明らかになった．特に，国・公益法人等および民間組織については，ソースコードを公開することに関して抵抗があると考えられる．開発言語については，演算部分については大半が Fortran であり，ついで C++がその半分程度の件数であった．

データの利用に関して，標準インターフェースに対応しているソフトウェアや，広く一般に公開されているデータをオンラインで取得できるソフトウェアはなかった．

2-4　水理・水文解析に関わる人材・組織の状況

水理・水文解析に関わる人材・組織の状況を，国土交通省の河川事務所等の河川技術者との定常的で数多くのコミュニケーションを元にまとめた．現在，我が国では国家公務員の定数が減少しつつあり（図-2.4）[15]，河川管理を所管する国土交通省の人員は経年的に減少してきている．その結果，職員が河川事業に関する計画検討等の解析にかける時間や

労力も圧縮しなければならない状況である．河川事務所では，複雑な数値計算を含むこれらの解析は一部の例外を除き，民間コンサルタントに外注しているが，発注者側の河川技術者は，業務の品質の確保のための労力を割くことが難しい状況になってきている．コンサルタントの業務報告書に記載されたソフトウェアの計算条件やそれを用いた解析方法をソフトウェア上で確認することができず，職員が解析結果の妥当性を対外的に説明するのが困難になることもある．また，同じ現場において類似した解析業務を外注することは多々あるが，外注先の民間コンサルタントが変わった場合，コンサルタント同士が同一の解析ソフトウェアを使っているわけではないので，同一のパラメータやデータセットを用いても同じ解析結果が得られるとは限らず，同じ結果が得られるように解析モデルを改良するのに多大な労力を要することがある．

河道断面データや水文データ等の河川・流域に関するデータの品質管理は管理主体である国土交通省の担当事務所と地方整備局等が行うものであるが，解析に必要なデータの品質を確保するための労力を割くのも難しい状況のため，その一部を外部に委託している．かつては，観測，照査，解析を同じ河川技術者が詳細を監理できたが，現在はそれができない状況にある．

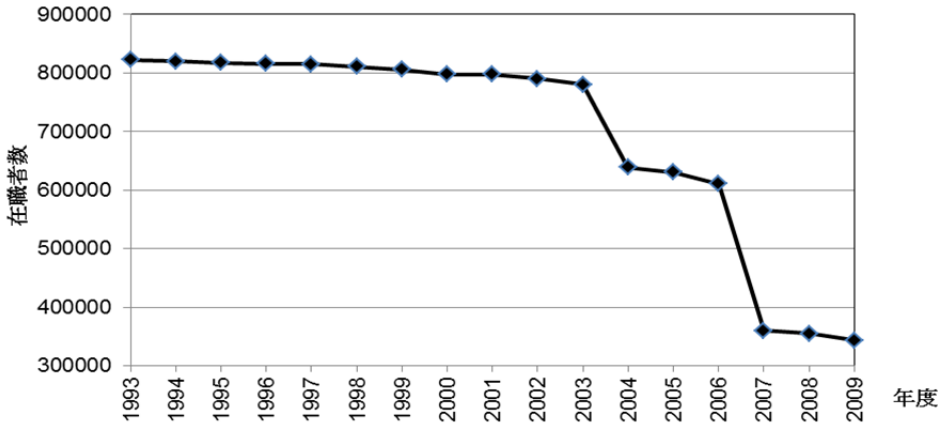


図-2.4　国家公務員の在職者数の推移 [15)]

2-5　国内事業における水理・水文解析の課題分析

河川・流域に関するデータの状況，水理・水文解析ソフトウェアの開発状況，および河川・流域に関する技術者の状況は，以下のように分析できる．

河川・流域のデータについては，データ収集や配布のための枠組みは整いつつある．データを提供するための標準インターフェースも整備されつつある．しかしながら，データ整備については，作業の進捗が国土交通省の担当事務所の努力にかかっており，人員が削

減されるなか，事務所が労力を割くのが難しい状況になってきている．

水理・水文解析ソフトウェアについては，河川・流域に関するデータの整備や配布のための枠組みが整いつつある一方で，それに対応してデータをオンラインで取得することができる解析ソフトウェアはなかった．解析ソフトウェアは，個々の研究者や組織がそれぞれ独自に開発しており，連携するための枠組みがなく，自分で開発したソフトウェアを自ら使っているか，特定目的のソフトウェアを特定の人が使っているという状況である．概してソースコードを公開するソフトウェアが少ないことも，それが解析モデル開発者同士の協力を妨げている可能性がある．

以上のように，解析モデル開発者同士が協力しにくい状況であり，解析結果等の資産をより多くの現場に活用するのは難しくなっている．また，大学や他の民間コンサルタントが現場等からの知見を得て，改良に繋げる機会が少なくなっている．この結果，「データ整備の進捗」，「解析ソフトウェアの改良」，「技術者の水理・水文解析能力の向上」のそれぞれの状況を改善していくのが難しい状況であると推察できる．これらを解決するためには，事業の現場でも，大学や民間コンサルタントにも使うことができる汎用プラットフォームを導入することが解決策の一つになると考えられる．

第 3 章　既存の汎用プラットフォームの開発状況 [7)]

3-1 概説

ここでは，新たに開発する水理・水文解析のための汎用プラットフォーム（以下,「汎用プラットフォーム」と呼ぶ）の開発方針や機能要件を検討するため，既往の汎用プラットフォームを調査することとした．既存の世界の主要な水理・水文解析ソフトウェアとしては，欧州の Mike シリーズ（デンマーク水理環境研究所・デンマーク）や InfoWorks シリーズ（HR Wallingford，英国），米国の HEC シリーズ（米国陸軍工兵隊水文工学研究所）等がある．これらのソフトウェアは，その中で流出モデル，河道モデル，氾濫モデル，海洋モデル等を含み,それ自体が水理･水文解析モデルのプラットフォームとしての機能がある．これらのソフトウェアは，この機能があるために，世界の主要なソフトウェアとなり得たと考えられる．しかしながら，これらのプラットフォームは，他者が解析モデルを開発することができないので，ここで呼ぶところの汎用プラットフォームの範疇には入れないこととする．

ここでは，そのプラットフォーム上で稼動する解析モデルの開発仕様をユーザに公開して，異なる開発者の解析モデルを連携・協働して稼働させることのできるプラットフォームとして，OpenMI（欧州），OMS（農政務省・米国）および OHyMoS（京都大学・日本）について，開発の経緯や開発仕様を調査し，国内の水理・水文解析をとりまく課題への対応の適応性について分析した．

3-2 OpenMI (Open Modeling Interface and Environment)

OpenMI は，欧州水枠組指令 [16)]（Water Framework Directive）により求められる流域の統合水管理に対応するため，異なる水の過程を表現する解析モデル同士を結合して流域全体のシミュレーションを実行するために開発された解析モデル同士のデータ交換のためのインターフェース（Open Modeling Interface）および OpenMI が定める仕様の解析モデルを開発・実行するためのツール群（Open Modeling Environment）である．初版の OpenMI は，欧州 7 カ国および 14 組織が参加し，EC（欧州委員会）資金による HarmonIT プロジェクト [17)]により開発された．現在では，OpenMI Association という非営利組織により開発が進められている．プラットフォーム上で稼働する解析モデル（以下,「要素モデル」と呼ぶ）の演算単位は，既存の Mike11 や InfoWorks 等のアプリケーション・ソフトウェアとなっており，要素モデルの独立性が高い設計となっている（図-3.1）．初版の OpenMI の主要な開発者は，デンマーク水理環境研究所や HR Wallingford，Delft 水理研究所であり，自らが開発した世界の主要な水理･水文解析ソフトウェアである Mike シリーズや InfoWorks シリーズを OpenMI に対応させている．

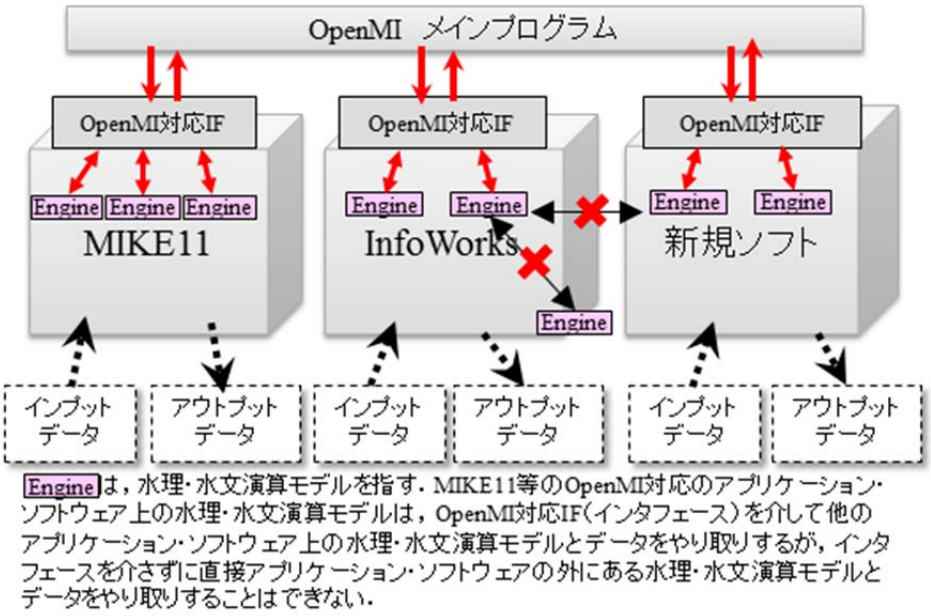


図-3.1　OpenMI

複数の要素モデルを連接するための設定は，Configuration Editor と呼ばれる GUI を有するエディタにより行う仕様となっており，Mike シリーズや InfoWork シリーズのユーザを意識していると考えられる（図-3.2）．

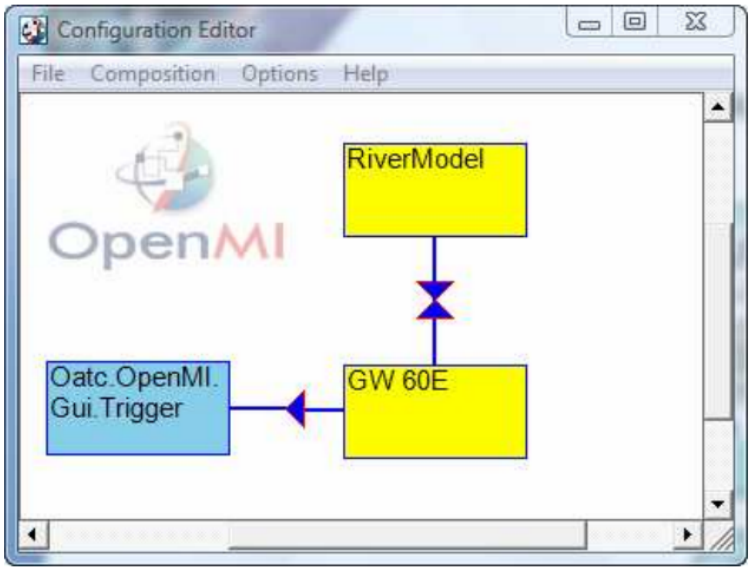


図-3.2　Configuration Editor

Open Modeling Environment は，オープンソースソフトウェアとして開発されており，仕様も公開されているので,技術的には誰でも OpenMI に対応したソフトウェアを開発することは可能である．Open Modeling Environment は，C#および Java で実装されているが，これらのプログラム言語以外の Fortran や C／C++等で記述された既存の解析モデルについても，OpenMI 対応に移植するは可能であるとしている．

OpenMI Association は最近，解析モデル間の通信仕様である OpenMI Interface Specification Ver. 2.0 を OGC（Open Geospatial Consortium）による国際標準化を目指している（2012 年 10 月現在）．

3-3 OMS（Object Modeling System）

OMS は，1996 年にドイツの Friedrich Schiller 大学によって，その原型が開発され，その後 2000 年に米国農政務省（USDA），米国地質調査所（USGS）を含んだ省庁間プロジェクトで開発が進められている[1)]．OMS は,Java の統合開発環境である NetBeans 上で実装され，オープンソースソフトウェアとして開発されている．OMS 上には，コンポーネント・エディタと呼ばれるエディタがあり，これを用いてコーディングすることにより，各解析モデルを結合して，シミュレーション・プロジェクトを作成する（図-3.3）．シミュレーション・プロジェクトの作成・実行だけでなく，GUI によるパラメータの入力や解析結果の可視化等のコンポーネントを含んだ統合環境となっている．NetBeans 上での開発言語は Java であるが,Fortran や C++で実装した解析モデルをラッピングすることにより使用することも可能としている．Colab[18)]（Collaborative Software Development Laboratory）と呼ばれる要素モデルの共同開発環境をウェブ上に設置し，開発コミュニティを形成している．

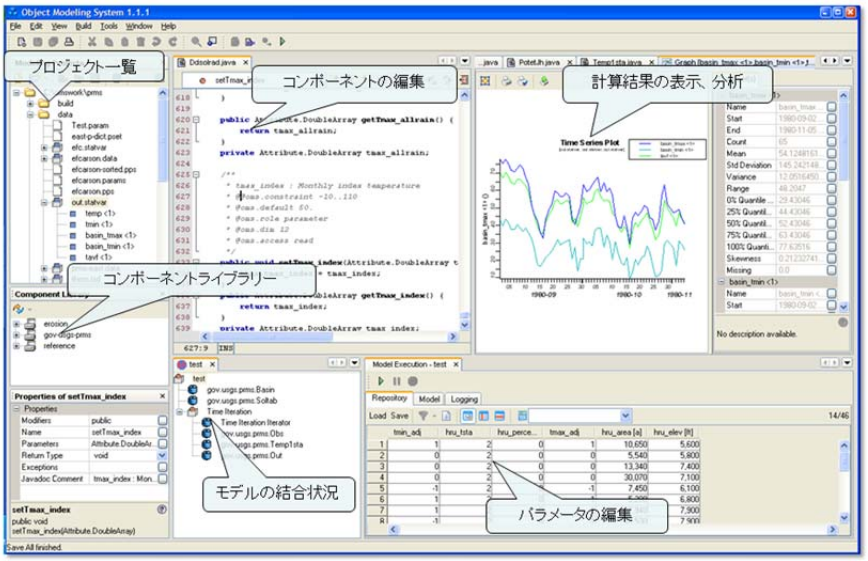


図-3.3　OMS のデスクトップ画面

図-3.4　Colab[18)]

3-4　OHyMoS（Object-oriented Hydrological Modeling System）

OHyMoS は，京都大学大学院工学研究科水文・水資源研究室により開発されている水文モデル構築システムである（図-3.5）．OHyMoS は，1990 年代初頭の C++等のオブジェクト指向のプログラミング言語が出始めたころに始まった．初版の OHyMoS は，C++により開発され，その後 Java および C#に移植されている．ソースコードは公開されている．

OHyMoS は，流域を斜面・貯水池・蒸発散・地下水流といった水文要素に分割し，それらに対応する要素モデルを相互関係に応じて再構築するという方法（構造的モデル化法）により流域の水文モデルを構築する．また，すべての要素モデルに共通した機能を有する要素モデルを基本型要素モデルとして定義し，オブジェクト指向プログラミングにおける継承機能を利用し,要素モデルの仕様の統一化および要素モデル作成の省力化を図っている．

OHyMoS は，GUI は持たずに，CUI（Character User Interface）により操作するソフトウェアある．モデル開発者は基本的には，シミュレーションを実施する際に，OHyMoS の本体システムのソースコードと要素モデルのソースコードを，要素モデルの接続状況を記載した構造定義ファイル一緒にコンパイルする必要がある．C#版は必ずしもソースコードは必要とせず，バイナリファイルのみでの提供も可能となっている．OHyMoS は，ソースコードが公開されており，ソースコード・レベルで改良可能なこと，要素モデルの接続状況等の

シミュレーション条件が汎用性の高いテキスト・ファイルで記述されていることから，汎用性の高いシステムと言うことできる．淀川実時間流量予測システム[19]の計算エンジンとして用いられ，リアルタイム・システムにも対応可能である．

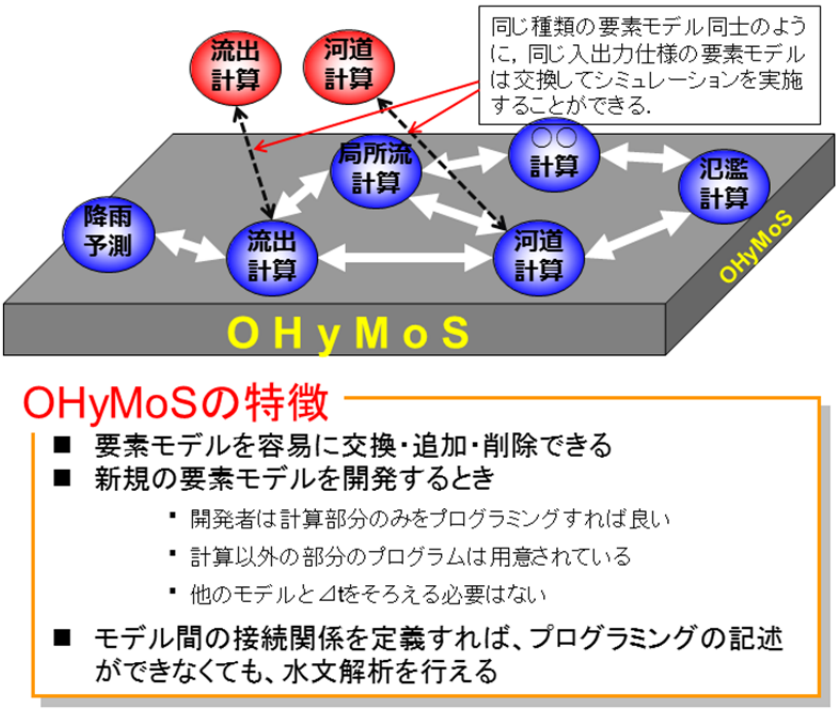


図-3.5　OHyMoS

3-5 既存の汎用プラットフォームの国内課題に対する適応性の分析

OpenMI は，世界の主要な商用のソフトウェア・メーカーが開発に加わっていることから，もともと商用のソフトウェアを稼働させることを前提として開発されていると考えられる．そのため，OpenMI では要素モデルのソースコードの提供を必要とせず，モデル開発者の権利保護を明確にしている．稼動する要素モデルは商用のソフトウェアであるので，当該要素モデルの改良による知見を他のモデル等に反映させることは難しいと言える．

OMS と OHyMoS の特徴は，類似している．両者とも要素モデルのソースコードの提供を前提としている．これにより，ソフトウェアの内部構造が明らかになる．このことにより，不特定の者が要素モデルを改良することが容易になり，知見の蓄積が図りやすい．OMS と OHyMoS が，研究機関が開発した知見の蓄積を重視した汎用プラットフォームであると考えると，適切な開発仕様の設定であると考えられる．ただし，要素モデルのソースコードの提供を前提としているので，物理的に要素モデルの内部構造を隠蔽する手段がなく，要素

モデル開発者の意図に反して，要素モデルに投入したノウハウが他者に流出することを防ぐことはできない．商用のソフトウェアが参画しにくい仕様といえる．

表-3.1に，既存汎用プラットフォームの国内の課題の適応性に関する属性を分析した結果を示す．国内の水理・水文解析をとりまく課題への対応策として，事業の現場でも，大学や民間コンサルタントにも同一の汎用プラットフォームを導入することが解決策の一つであることを考慮すると，当該汎用プラットフォームは，一般ユーザや研究者，モデル開発者といったそれぞれの属性のユーザのみを対象としたものではなくて，事業，業務，研究のそれぞれの現場のユーザすべてが使えるものでなければならない．また，民間コンサルタントの参入や知見の広がりやすさの両方を満足するものでなければならない．ここに挙げた３つの汎用プラットフォームのうち，これらすべてを満足するものはなかった．

表-3.1　既存の汎用プラットフォームの国内課題に対する適応性の分析

分析項目	OpenMI	OMS	OHyMoS
想定される対象ユーザ	商用ソフトのユーザ	研究者， モデル開発者	研究者， モデル開発者
モデル開発者の権利保護	保護しやすい	保護しにくい	保護しにくい
知見の広がりやすさ	広がりにくい	広がりやすい	広がりやすい

第 4 章　汎用プラットフォームの開発方針および機能要件の分析 [7)]

4-1　概説

新たに開発する汎用プラットフォームを「水・物質循環解析ソフトウェア共通プラットフォーム」(英語名称 CommonMP:Common Modeling Platform for water-material circulation)と名付け，国内における水理・水文解析における課題を踏まえた上で，開発目的を「水理・水文解析モデルの研究開発の促進」,「河川事業等への研究成果の反映の迅速化」および「水理・水文解析のアカウンタビリティの向上」とした.

そして，先述の既存汎用プラットフォームの分析結果を踏まえた上で，日本製の汎用プラットフォームである OHyMoS の機能を踏襲し,上記の開発目標を達成するための新たに開発する汎用プラットフォームの開発方針を「要素モデル開発の省力化・効率化」,「要素モデル開発者の権利保護」,「計算の透明性の確保」および「多様なユーザへの対応」と定めた．そして，これらの開発方針を実現するための主な機能要件を以下のように分析した.

4-2　要素モデル開発の省力化・効率化

CommonMP の開発においては，水理・水文解析モデルの研究開発を促進するために，要素モデルの研究者自身が要素モデルを開発することを前提として，GUI も含めた要素モデルの開発の省力化および効率化を図る方針とした.

4-2-1　演算要素の単位

CommonMP では，要素モデル開発者間の分業や協力が効率的に行われるように，水理・水文過程の基礎方程式や数値解法が同一であるブロックを一つの演算要素とした．例えば，洪水氾濫計算のために水理・水文モデルを利用する場合を考える．まず，対象流域を斜面流出，河道流下，氾濫浸水が発生する３つの場に分類する．次に，それぞれの場は，地形や土地被覆等の特性や観測地点や計算水文量を知りたい地点等の解析上の都合を考慮して，いくつかのブロックに分割される (図-4.1)．そして，ブロック間の関係は水の流れとそれを計算するために必要な情報の流れとして定義される．それぞれの場においては，各ブロックでの水理・水文計算のための基礎方程式や数値解法は同一であるが，ブロックにより異なるパラメータ値を与える方法がほとんどである．また，斜面，河道，氾濫浸水は異なる専門家がプログラミングし，これを連結して用いる場合が多い．連結後，斜面のモデルのみを別のモデルに置き換えて計算することもある．このような使い方に効率的に対応できるよう，基礎方程式や数値解法が同一であるブロックを一つの演算要素とした.

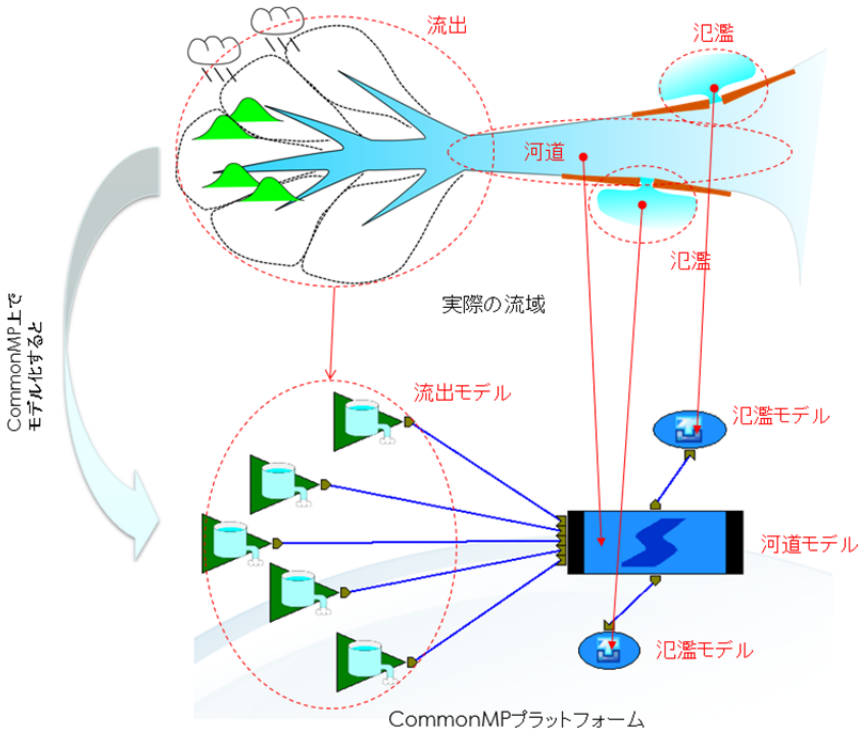


図-4.1 演算要素の単位による流域モデルの分割

このように演算要素の単位を基礎方程式や数値解法が同一であるブロックを最小単位にすれば，ある部分の基礎方程式や数値解法を改変した要素モデルを新規開発した後，新規開発モデルを入れ替えるだけで，インターフェースの設計に労力を割くことなく効率的に一連の計算ができることになる．一方，演算要素の単位を基礎方程式や数値解法が同一であるブロックを最小単位にすると，要素モデル間のデータのやりとりが複雑になり，演算時間が膨大になることが想定される．また，このことにより，より一つの要素モデルに接続される要素モデルの数が増え，陰形式でしか演算を進められなくなるケースが増加することが想定される．例えば，分派や合流のない単一河道を演算要素の単位とした場合，分派の接続を含むシミュレーション・プロジェクトにおいては，分派比を陽解法的に求めることはできないので，必ず演算要素間の収束演算が必要になる．一方，分合流を含む河道のネットワークを演算要素の単位とするように，より演算単位を拡大させた場合，河道分派比を求める収束演算は演算要素内で行えばよく，要素間収束演算は必要ない（図-4.2）．

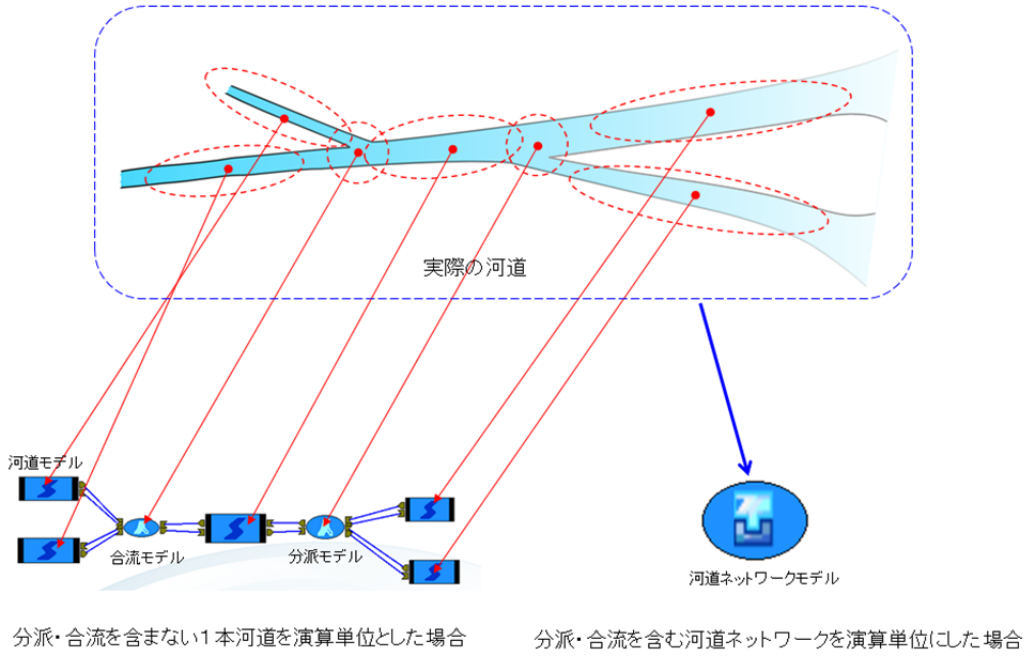


図-4.2　演算要素の単位

このように演算単位を拡大させた場合は，要素内収束演算で済むが，演算単位を細かくする程，要素間収束演算の必要性は高くなる．CommonMP では，原則として演算単位を基礎方程式や数値解法が同一である最小単位としたので，要素モデル間の収束計算ができることも機能要件とした．

演算単位を基礎方程式や数値解法が同一である最小単位とすると，同じシュミレーション・プロジェクトの中で同じ要素モデルを複数個使用する必要性が高くなる．配置された同一の要素モデルは，それぞれ独自のパラメータ設定や状態量，演算結果を持つことができないとほとんど意味がない．要素モデルを同じシミュレーション・プロジェクトの中で複数個配置し，それぞれの要素モデルが異なる状態量，パラメータ，演算結果を持てるようにするには，シミュレーションの実行時に複数個の要素モデルのオブジェクトのインスタンスを生成させる必要があり，そのような機能を個々の要素モデルに持たせる必要がある．この機能の実装は必ずしもオブジェクト指向プログラミングに精通しているとは限らない要素モデル開発者に重い負担を強いることとなるので，要素モデルの基底クラスにこの機能を持たせ，要素モデル開発者の負担を軽減させることとした．このように，CommonMP では，要素モデル開発者の負担を軽減させつつ，すべての要素モデルについて，同じ要素モデルを同一シミュレーション・プロジェクトで複数個使うことができるようにするとともに，パラメータ設定や状態量，演算結果を持てるようにした．

4-2-2　継承やひな形等の活用

CommonMP の要素モデル開発においては，要素モデル開発者のコーディング等の作業量を削減することで要素モデル開発の省力化を図ることとした．その１点目がオブジェクト指向プログラミング言語の機能である「継承」とプログラム・コードのひな形の活用である．オブジェクト指向のプログラム言語を用い，要素モデルに共通してもつ機能はプラットフォーム側で基底クラスを用意し，要素モデルを実装する際は，その基底クラスから継承する子クラスとして実装することとした．子クラスについても予めコーディングすべきところを記述したプログラム・ソースのひな形を用意した．これにより，モデル開発者は，ひな形から提供される部分については，ソースコードを実装する必要がなくなった．

基底クラスにおいては，計算手順（1 演算時間間隔分の演算処理，データ送信処理および計算を続けるかどうかの判断処理の順序，並びに計算を続けるかどうかの判断処理手続き）や隣接する要素モデルからのデータの受信に関する処理（演算実行時の時刻における入力値を求めるためのデータの補間処理等）が定義されている．子クラスにおいては，1 演算時間間隔分の演算処理およびデータ送信処理の手順をユーザが実装する．これにより，かなりのコーディング量を減らすことができる．

２点目は，シミュレーションで共通に用いられる機能は，プラットフォーム側で用意することにより，要素モデル開発の省力化を図ることである．CommonMP の演算要素モデルは，原則として直接降雨量や流量等の水理・水文データをファイル入出力しない仕様としている．データの入出力等のシミュレーションで共通に用いられる機能は，入力要素モデルおよび出力要素モデルとしてプラットフォーム側で用意する（図-4.3）．モデル開発者は，それらの機能を使うことができるので，データの入出力に関するコード量を減らすことができる．また，入出力要素モデルを用いると，異なるデータ・フォーマットへの対応は，入出力要素モデルを取り替えて対応すればよくなり，データ・フォーマットの変更が要素モデルの設計に影響が及ばなくなる．

図-4.3　入力要素モデルと出力要素モデル

4-2-3　統合開発環境の活用

統合開発環境（IDE: Integrated Development Environment）とは，プログラム作成に当たって，ソースコードの入力補完等のプログラムの編集・修正を支援するとともに，差分コンパイル（機械語翻訳），デバッグ，ファイル管理等を GUI から統合的に行うことのできるソフトウェアである（図-4.4）．現在，プログラミングにおいては，統合開発環境が多く用いられている．要素モデルの実装は，開発効率を考慮し，統合開発環境を利用することとする．また，幅広いユーザが要素モデル開発に参画できるように，統合開発環境は無償で提供されるものである必要がある．必要に応じて，既存の統合開発環境に機能拡張を施し，要素モデル開発を支援する機能を追加することとする．

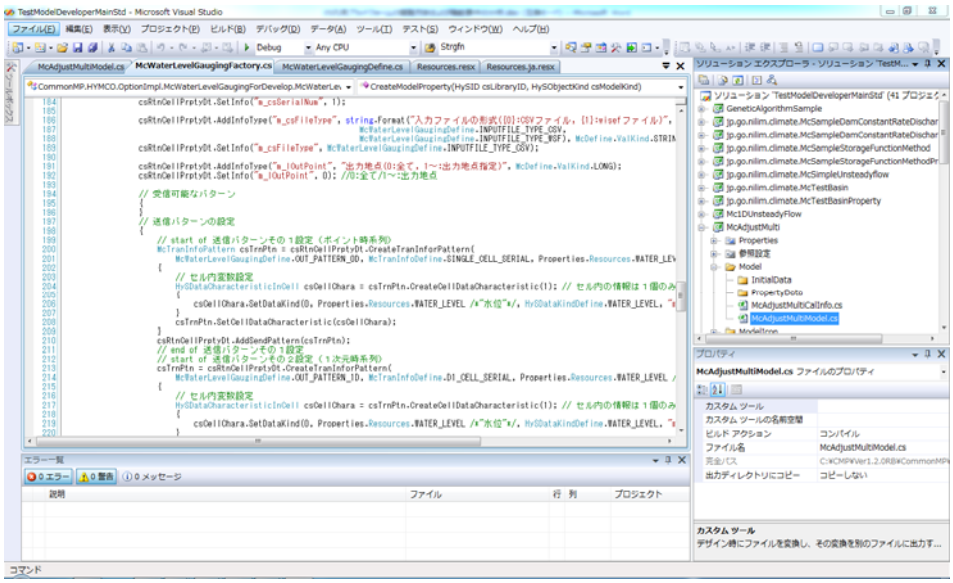


図-4.4　統合開発環境

4-2-4　要素モデルのパッケージ化

CommonMP の要素モデルは，複数のファイルから構成されることが考えられ，その場合 CommonMP の所定のディレクトリにそれぞれのファイルをコピーしたり，設定ファイルを編集したりする必要がでてくる可能性がある．CommonMP では，複数にまたがるファイルをひとまとまりにして（パッケージ化），そのまま CommonMP にインストールできたり，削除できたりできるようにする．要素モデルのパッケージ化をスムーズに行えるように，統合開発環境から要素モデルの配布パッケージを出力できるようにすることとする．要素モデルが流通しやすくなるように，この配布パッケージは，電子メール等で送付することもできるようにする．

4-2-5　過去のプログラム資産の活用

「2.3　水理・水文解析に用いるソフトウェアの状況」で示したとおり，既存の水理・水文解析ソフトウェアの数値演算部分は，Fortran で記述されたものが多い．既存のプログラムを有効活用し，重複開発を防ぐために，既存の Fortran 等の CommonMP の開発言語以外のプログラミング言語で開発された解析モデルについても利用できるようにすることで要素モデル開発の効率化に寄与することとする．CommonMP は，その機能要件からオブジェクト指向のプログラミング言語で開発されることが確実であり，Fortran で記述されたプログラムを CommonMP プラットフォームから直接制御することはできない． Fortran 等で開発された解析モデルは，インターフェースを介してプラットフォームから演算制御されることが想定されるが,そのインターフェースについても,汎用的なものとする必要がある．

4-3　要素モデル開発者の権利保護

「2.3　水理・水文解析に用いるソフトウェアの状況」で示したとおり，現在の日本製の水理･水文解析ソフトウェアは,ソースコードを開示していないものが多いと考えられる．特に民間組織がもつソフトウェアについては，その傾向が強いといえる．それは，モデル開発者がソースコードを開示することにより，ノウハウが流出することや，有償ソフトウェアとしての販売に支障をきたすことを恐れているからであると考えられる．CommonMP の活用においては，民間組織の参画も不可欠であるので，CommonMP の開発においては，モデル開発者の権利が保護されることを重視した．そのため，CommonMP の要素モデルは，ソースコードがなくても，実行形式のファイルさえあれば，シミュレーションを実行できるようにすることを機能要件とする.また,要素モデルの著作権者の権限を明確にするために，利用許諾条件（利用制限事項，複製，再配布の可否等）を設定できるようにする必要がある．モデル開発者の工夫により，コピープロテクトやユーザ認証を実装すれば，開発者の意図に反して開発した要素モデルが流通することを防ぐとともに，有償の要素モデルが開発されることにも対応できるようにする．

4-4　計算の透明性の確保

CommonMP は，河川事業等の公共事業で活用されることを前提としているので，アカウンタビリティ（説明責任）が強く求められる．そのため，CommonMP の開発においては，要素モデルの計算方法の透明性を確保することを重視した．4-3 で示したとおり，CommonMP の要素モデルは，ソースコードがなくても実行形式のファイルのみが提供されれば，当該要素モデルを用いたシミュレーションが実行できるようにするが，要素モデルの配布パッケージにはモデル開発者の判断によりソースコードも添付できる形式とする．また，要素モデルの計算方法等を示した解説書を要素モデルの配布パッケージに添付でき，ユーザは CommonMP 上でその解説書を閲覧できるようにする．このように CommonMP の要素モデルに

は，計算の透明性を確保することを支援するための仕組みを用意する．

4-5　多様なユーザへの対応

河川事業等への研究成果の反映の迅速化を実現するため，現場の河川技術者，河川コンサルタント等の解析を専門とする技術者，および水理・水文解析モデルの研究・開発者が同一の解析モデルおよび条件設定で水理・水文シミュレーションを実施できることが望ましい．そのため，CommonMP の開発においては，特定の事業・研究等を目的としたユーザではなく，幅広いユーザを対象として，それぞれの立場の者にとって使いやすい利用形態をとれるソフトウェアにするという開発方針をとった．

必ずしも水理・水文シミュレーションに精通していない事業現場の技術者にとって使いやすいように，GUI を備えたスタンドアロンのアプリケーション・ソフトウェアの形態で提供する．解析業務の定形作業を大量に行うユーザに対しては，CUI から実行できるようなソフトウェアの形態でも利用できるようにする．シミュレーション条件は，外部からテキスト・ファイルで与えることができるようにし，パラメータを変化させたシミュレーションを何ケースも繰り返し実行することも可能となるようにする．

さらに，将来的には洪水予測システムの計算エンジンとして，OHyMoS の淀川実時間流量予測システム[19]のようにサーバ運用することも想定している．

CommonMP は，多様なユーザを対象とすることから，どのような機能が必要であるかを想定することはできないので,ユーザが独自追加できるような拡張性を持たせる必要がある．CommonMP は，シミュレーションの実行だけではなく，データの取得やシミュレーション結果の可視化もプラットフォームの中でできるようにする．要素モデル以外のプラットフォームへの付加モジュールは，機能拡張ツールと呼ぶこととするが，この部分についても，ユーザが独自に開発できるようにする．

機能拡張ツールの一つとして，GIS（地理情報システム）を可視化ツールとして導入することとし，データの一元管理，地図上で氾濫水位の表示等ができるようにする．地図上に表示できる情報については，GIS に集約することによって一元的に河川・流域管理ができるようにする．

2012 年 6 月には，水文水質データ取得ツールが公表されている．このツールは，「2-1 河川・流域に関するデータの整備状況」で述べた標準インターフェースを用いて水文水質データベースからリアルタイムデータや過去データを取得するものである．今後，データベース接続については，増えていくものと考えられる．

第5章　プログラムの設計・実装[7)]

5-1 概説

第4章で述べた汎用プラットフォームの機能要件を実現できるように，本章ではプログラムの設計を行った．第4章で述べた機能要件から，CommonMPはオブジェクト指向のプログラミング言語で開発する必要がある．まず，この条件を踏まえた上で，想定されるユーザや要素モデル開発者の属性を考慮して，CommonMPの開発言語，開発環境，実行環境等を設定した．

次に，機能要件を満たすようなシミュレーション・プロジェクトの演算制御方式と要素モデルの演算の進め方に関するタイプ，要素モデル間収束演算，要素モデル間の伝送情報および要素モデル内部での演算処理を実現するための要素モデルのクラス設計と各クラスが持つ処理のための設計を行った．

この他に,要素モデルのFortranプログラムのためのラッピング手法,機能拡張ツール,及び複数シミュレーション・プロジェクトの同時並行稼働（マルチプロジェクト演算）機能の設計を行った．Fortran プログラムのためのラッピング手法については，２種類の方法について実装し，性能試験を行った．機能拡張ツールについては，水文水質データベース[8)]から水文データを取得するためのツール（水文水質データ取得ツール）を設計し，実装した．マルチプロジェクト演算機能については，水文水質データ取得ツールのリアルタイムデータ取得機能を使った卓上簡易洪水予測システムを作成し，性能確認のために試験を行った．

5-2　開発言語・開発環境の選定

まず，CommonMPは，その機能要件からGUIを持つソフトウェアであり，それをある程度短期間で開発を進めなければならないという条件がある．ソフトウェア開発においては，OS（オペレーション・システム）上で直接開発するか，何らかのミドルウェアを介して開発する方法がある．OS上で直接開発する方法は，出来上がったソフトウェアの実行速度が速い反面，プログラミングが複雑で開発効率が悪いという欠点がある．また，異なるOSでソフトウェアを実装する場合，OSごとに実装を行わなければならないので，異なるOSでソフトウェアを開発する場合は,労力が大きくなる.ミドルウェアを介した開発は,その逆で,ソフトウェアの実行速度は遅いが，プログラムの開発は比較的容易で，異なるOS上で開発する場合も,ミドルウェアがその差を吸収するので,原則として同じコードを一度書けば,そのミドルウェアが対応したOS上で動かすことができる.

CommonMPの開発は，その機能要件から河川事務所等の技術者のパーソナル・コンピュータ上で動かすことを想定しており，大規模計算を現段階においては想定していない．よってCommonMPは，開発効率を優先して，何らかのミドルウェア上で開発することとする．

CommonMPのミドルウェア（開発環境）の候補として，まずWindows上で動くことと，GUI付きのソフトウェアが開発でき，ある程度普及していること要件として，.NET Framework，JRE(Java Runtime Environment)およびQtを取り上げ，それらの優位性を比較する(表-5.1)．

表-5.1　開発に用いるミドルウェア等の比較

比較項目	.NET Framework	JRE	Qt※1
概要	Microsoft社が開発したアプリケーションの開発・実行環境	ORACLE社が開発しているプログラム言語Javaの実行環境	C++で実装されたクロスプラットフォームの開発環境．ノルウェーのQt Development Framework社により開発されている．
対応OS	Windowsのみ	Windows，Mac OS X，Linux等	Windows，Mac OS X，Linux等
開発言語	C#，VB，C++等	Java	C++
開発環境	Visual Studio 高機能な有償版の他に機能を限定した無償版がある．	JavaSDK Eclipse，NetBeans等 無償の開発環境がある．	Qt Creator 商用版のほか，GPL，LGPL※2の無償版あり
長　所	各種ライブラリが充実している．Windows専用の開発ツールであり，GUIを含めて開発効率が高い．	各種ライブラリが充実している．開発コミュニティの層が厚いので，将来的な開発が続くことが期待できる．並列化ライブラリを用いれば数値計算が速くなることが期待される．	ミドルウェアを必要としないので，実行速度が速い．
短　所	開発がMicrosoft社のみに依存しており，将来的に開発が続くかが不明． 対応しているOSがWindowsのみ．	Windows上では，実行速度が遅いと言われる． GUIを用いたプログラムでは開発効率が低い．	日本語の開発情報が少ない． GPLの開発環境で開発されたソフトウェアにはGPLが適応される． 商用版の開発環境とGPL，LGPLの開発環境を同一プロジェクトの中で使うことはできない． 商用版の開発環境は高価

※1 Qtは，開発環境から直接それぞれのOS上で動く実行形式ファイルを作成することができるので，ミドルウェアを必要としない．
※2 Lesser General Public Licenseの略，オープンソースソフトウェアのライセンスの１形態．

.NetFrameworkは，プログラムの開発効率が高く，無償の開発環境を使うことができる点が優れている．だだし，開発をMicrosoft社１社に依存しているのと，実行環境がWindowsのみに限定されるのが難点である．

Qtは，開発されたソフトウェアが高性能であること，様々なOS上でのソフトウェア開発

を１種類のソースコードで管理できることが優れている．しかしながら，日本語の開発情報が少ないので，開発環境に習熟するのに労力を要することが想定される．また，商用版の開発環境が高価（Qt Desktop Full Framework Single Targetの１年間のライセンス価格が約50万円）であることと，無償版の開発環境は，開発されたソフトウェアのライセンス上の制限があり，要素モデル開発者が導入するのは難しいと考えられる．

JRE（Java Runtime Environment）は，開発コミュニティの層が厚く，開発情報を入手しやすく，今後の開発が継続されることが期待できることが優れている．しかしながら，GUIを持つソフトウェアを効率的に開発することが難しいという難点がある．

現在の河川事務所等ではパーソナル･コンピュータのOSとしてWindowsが大部分を占めていることを考慮すると，OSがWindowsに限定されるが，開発効率等が優れている.NetFrameworkを選定するのが適当と考えられる．将来的に，洪水予測システムのようにサーバ運用することや並列化を伴う大規模計算をするような場合には，サーバの主要なOSであるLinuxに対応することが必要になってくることが想定される．

開発言語については，.NetFramework上では，主にC#，Visual Basic，C++等を使うことができる．そのうち，C#は.NetFrameworkの開発時に.NetFrameworkで使うために作られた言語であり，.NetFramework上では最も開発効率が高いプログラミング言語である．Visual Basicは，最もユーザの多いプログラミング言語の一つであるが，.NetFrameworkが開発される前から存在したプログラミング言語であり，.NetFrameworkを対応させるように言語仕様を拡張させたため，わかりにくくなったところがある．新規に学習するとすれば，Visual BasicよりもC#の方が適していると考えられることと，C#は，JavaやC++と文法が類似しており，これらのプログラミング言語を経験したことのある人にとっては馴染みやすいことから，CommonMPの開発言語はC#にすることとする．

5-3　演算制御方式と要素モデルの演算実行方法等の設計

CommonMPの要素モデル開発においては，モデル開発者が，例えば斜面流出，河道流下，氾濫浸水等のそれぞれの専門研究者による新規開発モデルを，最小の労力で連結して計算できるように，演算要素の単位を基礎方程式や数値解法が同一である最小の演算要素の単位とする開発方針をとっている．このため，要素間収束演算の必要性が高くなったため，要素間収束演算ができることを機能要件とした．これらの機能要件を実現するために，要素モデル間のデータのやり取りに関する演算制御方式および個々の要素モデルの演算の進め方に関するタイプ等を下記のように設計した．

CommonMPの演算制御方式は，水文系の降雨流出解析を主眼として開発されたOHyMoSの機能を参考に設計した．降雨流出解析においては，上流からの要素モデルで算出された流量や降雨量等の水文量を下流の要素モデルの入力値として渡すことにより演算が進められ，要素モデル間の演算順序を上流から下流に一意に定めることができる（図-5.1）．

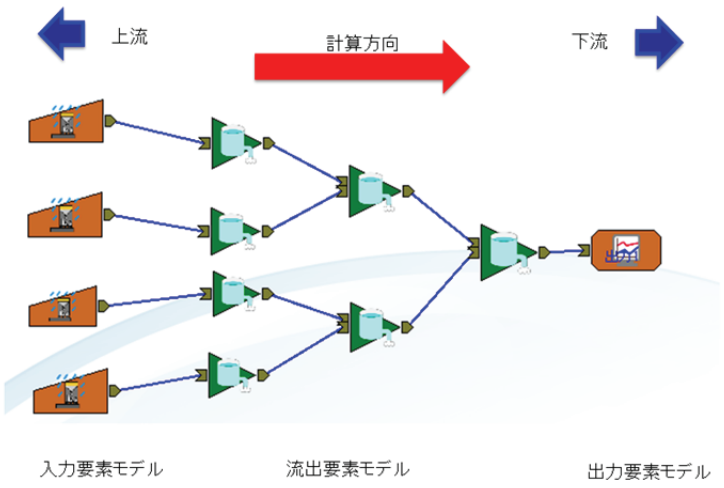


図-5.1　非同期型演算制御に適したシミュレーション・プロジェクト

また，流出モデルは，当該時刻における入力値に対して，同一時刻における出力値を算出する演算モデルが多い．CommonMPでは，要素モデル間の演算順序を一意に定めることのできる演算制御方式を後述の同期型演算制御に対して非同期型演算制御と呼ぶこととし，当該時刻における入力値に対して，同一時刻における出力値を算出する要素モデルを現状計算型要素モデルと呼ぶこととした．そして，この演算制御方式と要素モデルのタイプの組み合わせを基本とした．

一方，要素モデル間の接続がループを構成したり，双方向接続があったりすると，要素モデル間の演算順序を定めることができず,各要素モデル同士が1演算時間間隔ごとに同期をとりながら演算を進める必要がある（図-5.2）．この演算制御方式を同期型演算制御と呼ぶこととした．

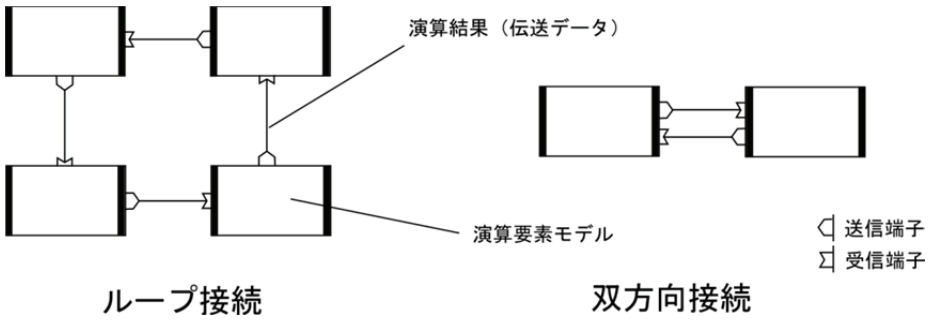


図-5.2　ループ接続と双方向接続

河道不定流モデルのように，当該時刻における入力値に対して，1演算時間間隔後の出力値を算出するタイプの要素モデルを未来計算型要素モデルと呼ぶこととした．CommonMPでは，非同期型演算制御と現状計算型要素モデルによる水文系のシミュレーションを基本として，河道不定流モデルのネットワークを構成するシミュレーションのような水理系のシミュレーションのために，演算制御方式として同期型演算制御，および演算の進め方に関する要素モデルのタイプとして未来計算型要素モデルを加えることにより，機能拡張したと言うことができる．また，組み合わせ可能な演算制御方式と要素モデルのタイプは，表-5.2のとおりとなる．CommonMPでは，要素モデル間収束演算ができることも機能要件にしたが，要素モデル間収束演算の設計は，演算制御方式や要素モデルの演算の進め方に関する設計とも密接に関連するので，以下に演算制御方式と要素モデルのタイプ，および要素モデル間収束演算の設計について述べる．

表-5.2　演算制御方式と要素モデルのタイプの組合せ

演算制御方式 / 要素モデルのタイプ	非同期型演算制御	同期型演算制御
現状計算型要素モデル	可能	不可能
未来計算型要素モデル	可能	可能※

※ この組合せのみループ接続や双方向接続，要素間収束演算が可能である．

5-3-1　演算制御方式の設計

CommonMPにおける演算制御方式については，要素モデル間の演算順序の違いにより，非同期型演算制御と同期型演算制御の２種類が用意されている．要素モデル間の演算順序が上流から下流に向かって順次計算を進める方式を非同期型演算制御，各要素モデルが1演算時間間隔ごとに同期をとりながら演算を進める制御方式を同期型演算制御と呼んでいる．以下に，非同期型演算制御と同期型演算制御の設計について述べる．

5-3-1-1　非同期型演算制御

非同期型演算制御は，要素モデル同士の接続関係において，上流（演算結果の出力情報の送信側）と下流（演算結果の出力の受信側）が定義でき，演算順序が特定できるシミュレーションおいて，順次上流の要素モデルから演算を実施する制御方法である．図-5.3において，まず要素モデルAがプラットフォームが設定した中間目標時刻を超えるまで独自の演算時間間隔で演算を行い，その結果はすべて時系列のデータとして要素モデルBとの

間の伝送線路に蓄積される．要素モデルBは，そのデータを自らの演算時間間隔に合うように内挿補間した上で受け取り，その補間データを用いて，中間目標時刻まで独自の演算時間間隔で演算を行い，結果を下流の要モデルCとの間の伝送線路に蓄積する．中間目標時刻に達して，演算結果を要素モデルBが受け取った段階で，要素モデルAが伝送線路上に蓄積したデータは解放される．このようにして，データ蓄積による計算機メモリの不足を防いでいる．すべての要素モデルの演算が終了したら，次の中間目標時刻に向かって要素モデルAから演算を行い，順次要素モデルB,要素モデルCが演算を行うという手順を繰り返す．中間目標時刻の時間間隔を長くすると，送線路上に蓄積するデータ量が多くなり，計算機メモリの使用量が増加する．一方，中間目標時刻の時間間隔を短くすると，データの転送が頻繁に行われ，演算速度が低下する．非同期型演算制御においては，要素モデルの演算順序が特定される必要があるため，要素モデルの接続に双方向接続やループ接続がないことが前提になる．この演算制御方式は，上流から順次流量が下流に伝わる水文モデル（流出モデル）等で用いられる演算制御である．

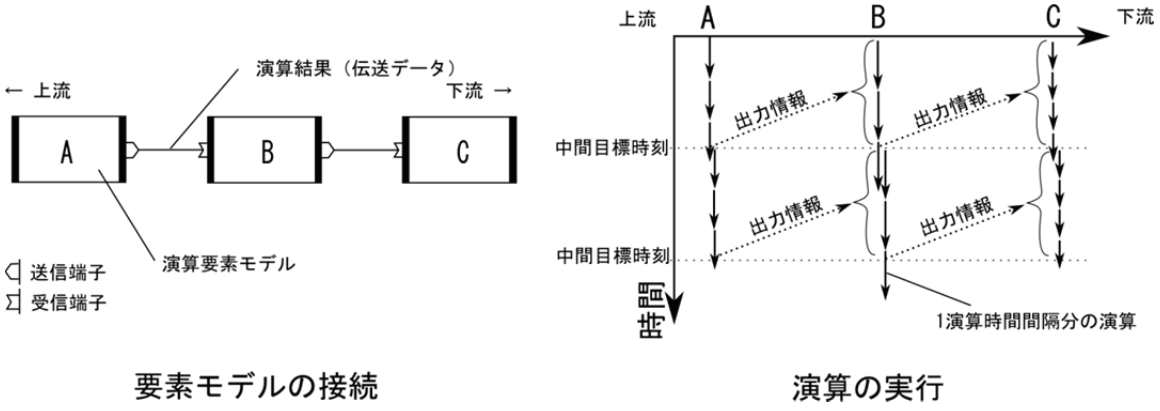


図-5.3　非同期型演算制御

5-3-1-2　同期型演算制御

同期型演算制御は,プラットフォームが設定した1演算時間間隔ごとに同期して各要素モデルが一斉に演算を行う演算制御方式である．それぞれの要素モデルが個別に中間目標時刻（開始時刻+⊿T）を超えるまで，独自の演算時間間隔で一斉に演算を実施し，演算結果の時系列データを接続先の伝送線路に蓄積する．図-5.4において，要素モデルA，B，Cがそれぞれ中間目標時刻まで一斉に演算を実施し，それぞれ接続先の伝送線路に演算結果のデータを蓄積する．中間演算時刻に達したら，各要素モデルは，当該要素モデルが必要とする時刻における伝送情報を内挿補間した上で受け取り，次の中間目標時刻まで演算を実施

する．以下，最終目標時刻までこの手順で演算を続ける．同期型演算制御においては，後述の未来計算型要素モデルを用いる．中間目標時刻間の時間間隔⊿Tは，各要素モデルの演算時間間隔よりも小さくなるように設定される．そのようにすれば，中間目標時刻に達した時に，必ず各要素モデルは１回演算を実施していることになり，各要素モデルは次の演算時間間隔の演算を行う場合に，必要な時刻の入力情報を隣接する要素モデルから取得できる．同期型演算制御方式では，要素モデルの演算順序を定める必要がないので，双方向接続やループ接続も可能となる．この演算制御方式は，双方向接続が必要となる水理モデル（たとえば，水位データは下流から上流に，流量データは上流から下流に伝える）に使われる制御方式である．

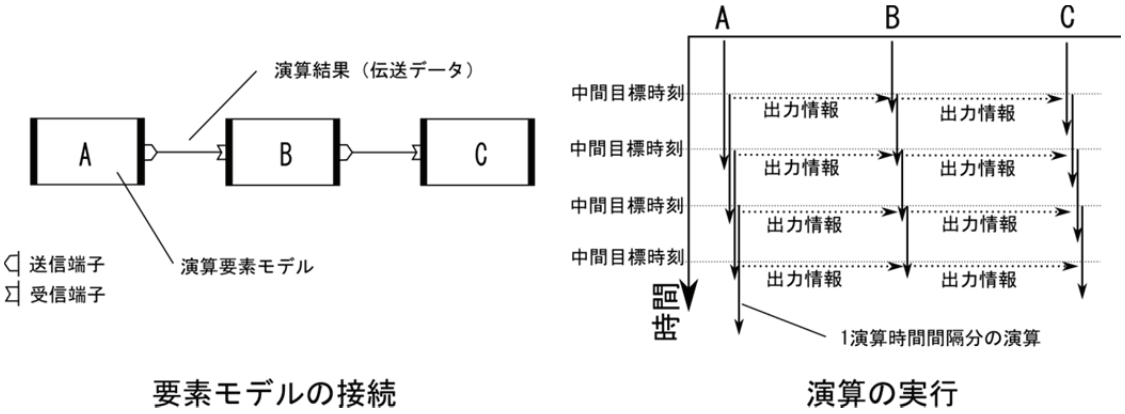


図-5.4　同期型演算制御

5-3-2　要素モデルの演算の進め方に関する設計

CommonMPの要素モデルは，演算の進め方の違いにより，２つのタイプに分類され，現状計算型要素モデルと未来計算型要素モデルが基底クラスとして定義されている.ユーザは，どちらかのタイプの要素モデルの基底クラスを継承して，要素モデルを実装する．

5-3-2-1　現状計算型要素モデル

現状計算型要素モデルは，現在時刻の入力情報を受け取り時刻を進めることなく演算を行い，演算結果を出力するタイプの要素モデルである（図-5.5）．不等流計算のような定常計算で用いられる要素モデルである．このタイプの要素モデルは，双方向接続やループ接続を含むシミュレーション・プロジェクトの中では利用することはできない仕様となっている．双方向接続やループ接続を含むシミュレーション・プロジェクトでは，要素モデル間の計算順序を定めることができないので，各要素モデルを一斉に演算させる同期型演算

制御にする必要がある．各要素モデルを一斉に演算させる場合，各要素モデルが演算を開始するときに，必ずしも隣接する要素モデルが独自に演算を進めて，演算に必要となる入力データを供給できる保証がないので，現状計算型要素モデルは同期型演算制御では利用できない仕様としている（例えば，同期型演算制御において，双方向接続を介して現状計算型要素モデルがある場合，演算を開始するときに互いに双方向接続先の現状計算型要素モデルから入力データを供給されることが必要となり，互いに入力データの供給を待つ状態（デッドロック状態）となり，演算を開始することができない.）．また，仮に最初の演算時間間隔における演算を実施できたとしたとしても，現状計算型要素モデルは未来の時刻における出力値を算出できないので，次の中間目標時刻において隣接する要素モデルに演算に必要な入力データを供給することができず，デッドロックが発生して演算を継続させることができなくなる．

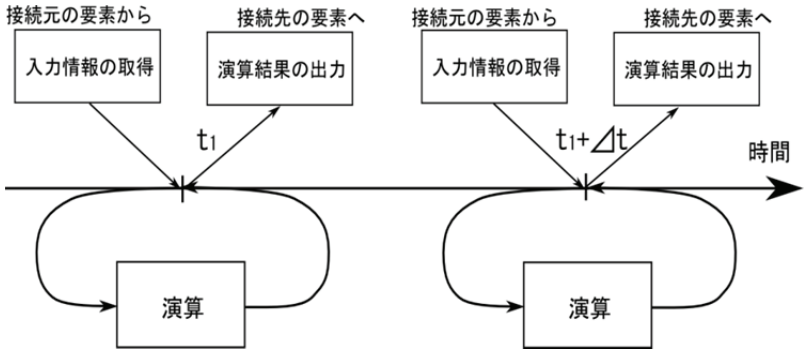


図-5.5　現状計算型要素モデルの計算方法

5-3-2-2　未来計算型要素モデル

未来計算型要素モデルは，現在時刻の入力情報を用いて，1演算時間間隔分演算を行い，1演算時間間隔だけ時刻を進めて演算結果を出力するタイプの要素モデルである（図-5.6）．このタイプの要素モデルは，不定流計算のような非定常計算で用いられる要素モデルである．このタイプの要素モデルは，双方向接続やループ接続を含むシミュレーション・プロジェクトの中で使用することができる仕様としている．また，未来計算型要素モデルは非同期型演算制御，同期型演算制御においても使用できる．各要素モデルを一斉に演算させる同期型演算制御の場合，シミュレーション開始時に各要素モデルが演算を開始させることができれば，各要素モデルは，1演算時間間隔分演算を進めて，隣接する要素モデルに入力データを供給することができ，次の演算時間間隔分以降演算を継続させることができる（ただし，それぞれの要素モデルは，演算を開始するときには，接続先からの入力データ

を必要としない仕様とする必要がある．そうしないと現状計算型要素モデルと同様にデッドロックが発生して，演算を開始できない.)．

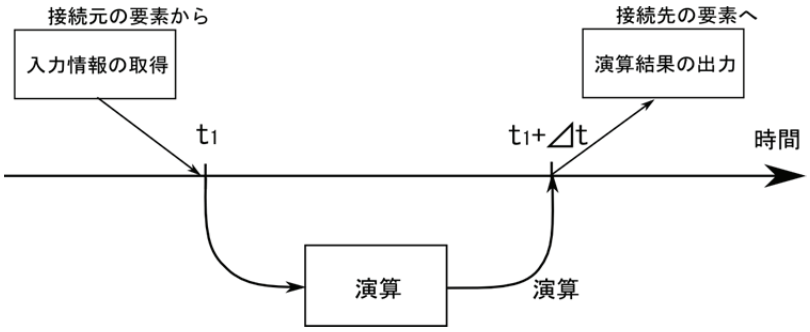


図-5.6　未来計算型要素モデルの計算方法

5-3-3　要素モデル間収束演算

同期型演算制御方式で双方向接続やループ接続した場合には，陽形式で演算できるとは限らず，複数の要素モデル間で収束計算を要するケースが出てくることが想定される．例えば，河道が分派するモデルの場合，分派比は上流の河道モデルからの流量と下流の派川モデルの上流端の水位が整合するように，収束計算を行い，分派比を調節する必要がある（図-5.7)．

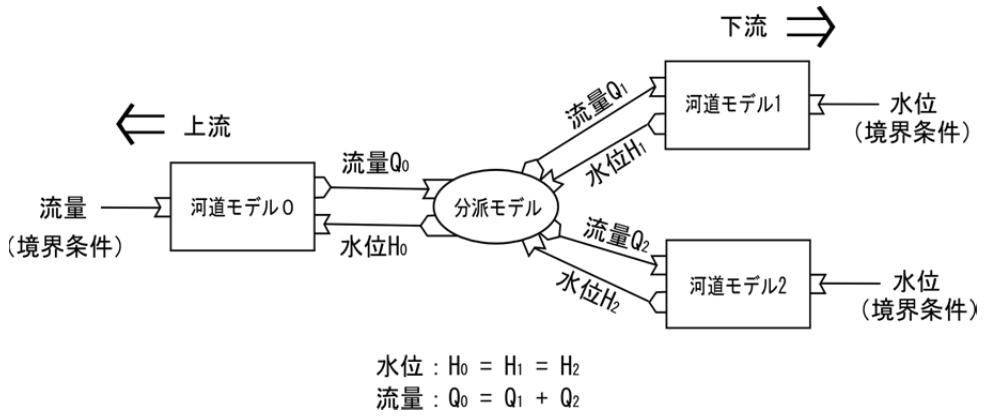


図-5.7　河道分派のため収束条件

そのため，CommonMPの同期型制御方式においては，要素モデル間の収束計算がサポートされている．要素モデル間で収束計算が必要なものをまとめて「収束演算グループ要素」と定義した（図-5.8）．収束演算グループ要素は，当該グループの外からは，一つの演算要素モデルのように見える．また，収束演算グループ要素とその外側の要素モデルを接続するため，通常の送信端子や受信端子とは異なる働きをする「中間端子」を定義した．収束演算グループ要素内の要素モデルは，ユーザが定義した収束条件に従い収束計算を行う．収束演算グループ要素は，中間端子を介して外側の要素モデルと伝送情報を交換することとなっており，収束計算をはじめる前に中間端子とつながっている入力端子から伝送情報を受け取り，収束計算が完了すると，中間端子とつながっている送信端子から伝送情報を出力する．

要素モデル間収束演算機能については，収束演算の枠組みはプラットフォーム側で用意し，初期値の与え方，近似解の与え方および収束判定をユーザが実装するようにした．具体的には，ユーザは収束演算に関する各要素モデルのメンバー変数を保有するクラス（収束演算用演算中データクラス）を，プラットフォームが提供する基底クラスから継承することにより実装するとともに，収束演算を制御するクラス（収束演算系制御クラス）の中の収束判定条件の初期設定（初期値を与える等）を行うメソッド，収束判定を行うメソッド，収束判定条件の再設定（近似解を与える等）メソッドの３つのメソッドを実装すればよい．

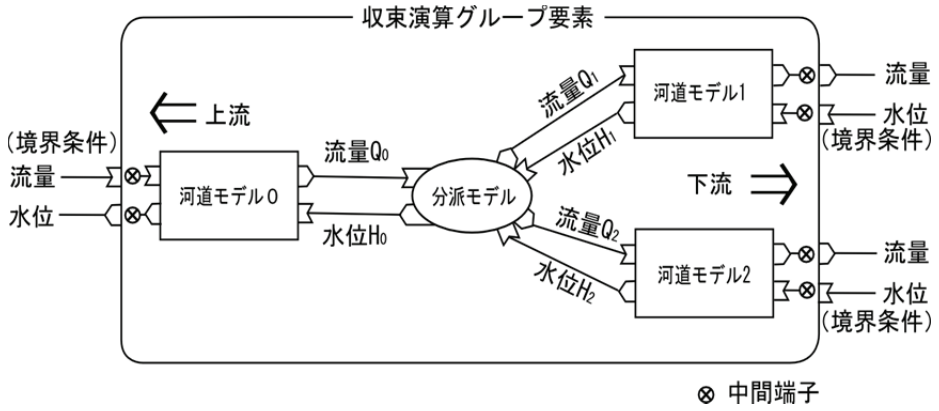


図-5.8　収束演算グループ要素

5-4　要素モデル間の伝送データ

CommonMPでは，シミュレーション実行時に要素モデル間で伝送される水理・水文量等のデータは，プラットフォームで伝送仕様を定め用意している．この他の演算要素モデルが入力要素モデル等を介さずに直接利用する河道断面データ等の境界条件に関するデータに

ついては，仕様は決められていない．以下に要素モデル間の伝送データの仕様の設計について述べる．

5-4-1　伝送データの構造

CommonMPでは，シミュレーション実行時に要素モデル間で，時系列の水理・水文量等を送受信しながら演算を進める．データは時刻（タイムスタンプ）と幾つかのデータをひとまとまりにしたデータセルという単位で取り扱う（セル型伝送データ）．例えば，図-5.9に示したセル型伝送データは，データセルの中に流量，水位，流速という３つの物理量を持ち，2012年12月11日10:15というタイムスタンプのついた１地点におけるデータを表す時系列情報である．これらの３つの物理量は，インデックスにより指定することができる．タイムスタンプは，CommonMPプラットフォームにより自動的に付与される．

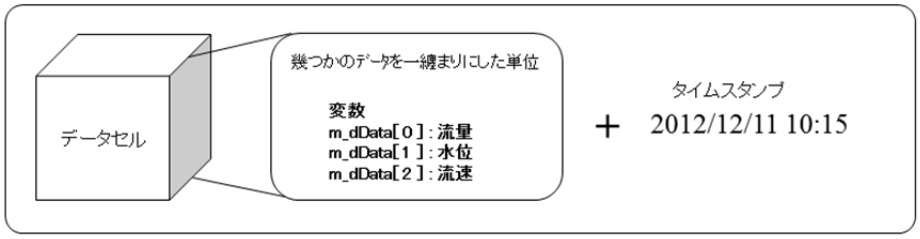


図-5.9　セル型伝送データ

5-4-2　伝送データの種類

データセルは，空間的な広がり（次元）を持つことができ，単独（０次元）のものから，長さ方向の次元を持つ１次元セル，平面的な広がりを持つ２次元セル，立体的な広がりを持つ３次元セルがあり，それぞれのセルは次元方向のインデックスにより指定することができる．また，これらの他にデータセルの緯度経度情報を持つGIS2次元メッシュ時系列情報とGIS3次元メッシュ時系列情報がある（図-5.10）．時系列情報の他に，データセルだけで構成され，タイムスタンプを持たない定常情報もある．

１次元セルは，例えば流下方向の次元を持つ１次元河道不定流モデルに用いることが考えられる．それぞれのデータセルに河道の各断面の水理量（水位，流速等）を格納することができる．２次元セルは，平面２次元氾濫モデルに用いることが考えられる．それぞれのデータセルに氾濫原のそれぞれのメッシュの水理量（浸水深，流速ベクトル等）を格納することができる．同様にして，３次元セルは例えば高さ方向の次元を考慮する湖沼モデルに用いることが考えられる．各データセルには各格子の物理量（流速ベクトル，水温，濃度等）を格納することができる．

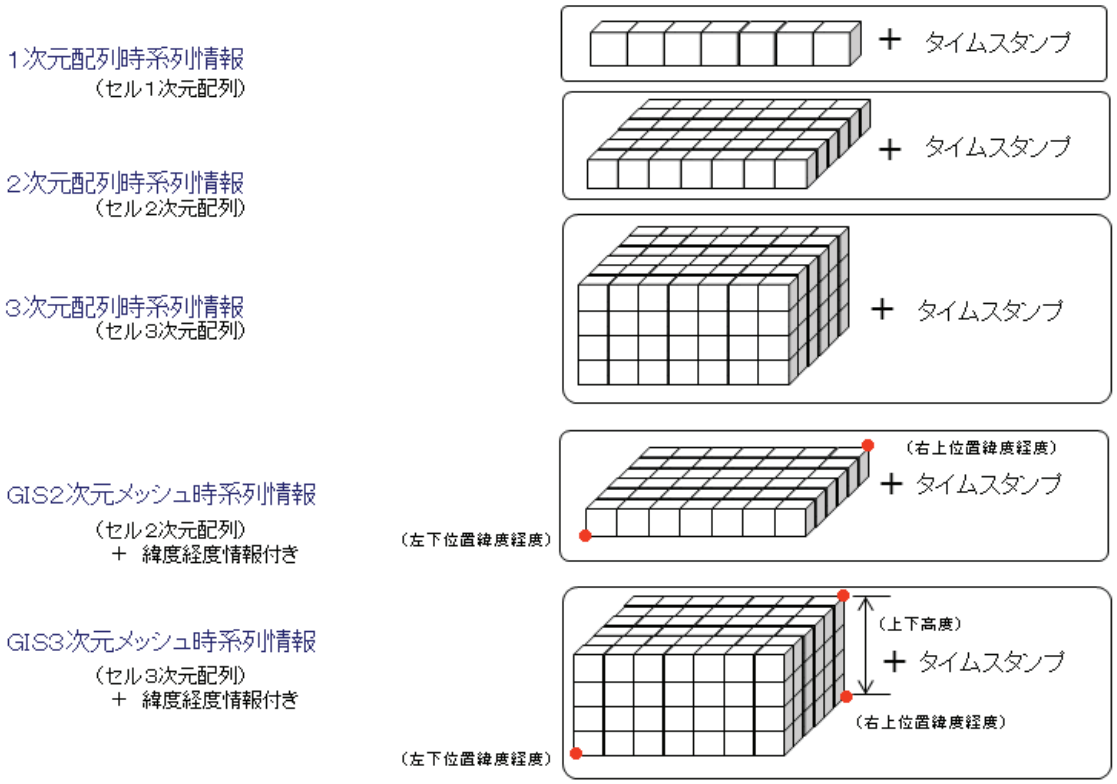


図-5.10　セル型伝送データの種類

5-4-3　伝送データのマッピング

データセルの個々のデータをどのような並びにするかは要素モデル開発者が任意に設定することができるが，正しい演算が行われるためには，送信側の要素モデルの伝送情報のセル内データの配列を知った上で，適切に受け取る必要がある．CommonMPのプラットフォームは，要素モデルのデータセル内のデータの並びを表示する機能があるので，ユーザは送信側のデータの配列を確認した上で，受信側の要素モデルが受け取るべきデータを手動で設定（マッピング）する（図-5.11）．

CommonMPの要素モデルの接続では，次元数の異なる伝送情報同士を接続することができなかったり，１種類しかデータを持たない伝送情報についても設定作業なければならない等（１種類しかデータを持たない伝送情報については，データの選択の余地がない），不便なところがあった．これについては，CommonMP Ver1.2より改善されている．前者については，プラットフォーム側で次元変換要素モデルを用意し，後者については，デフォルトで選択されるべきデータの種類を要素モデル開発者が設定できるようにしている．

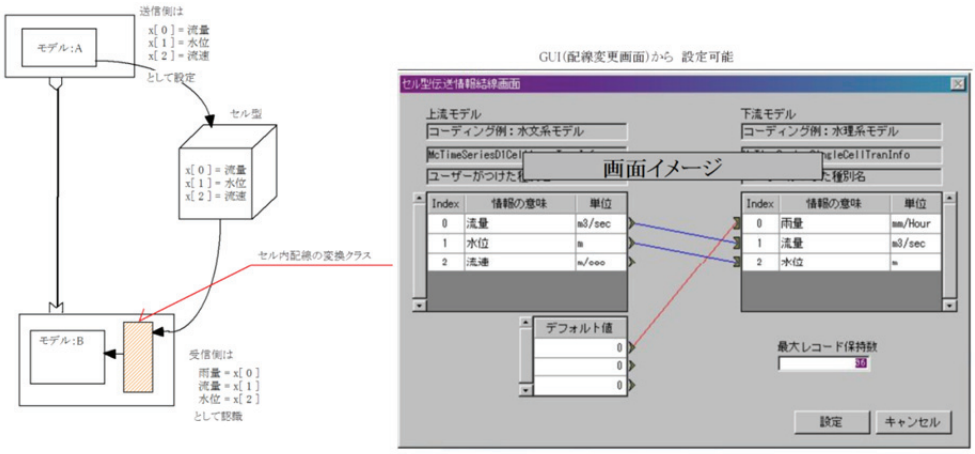


図-5.11　伝送データのマッピング

5-5　要素モデルの設計

「5-3-2　要素モデルの演算の進め方に関するタイプの設定」では，シミュレーション・プロジェクトの演算実行制御に関連して外部的にどのように挙動するかを中心に述べたが，ここでは要素モデルがシミュレーション・プロジェクトの中で内部的にどのように演算処理を実施しているかについて述べる．

5-5-1要素モデルを構成するクラス

CommonMPは，オブジェクト指向のプログラミング言語で開発されており，要素モデルもオブジェクト指向プログラムの構成単位であるクラスから構成されている．CommonMPの機能要件として，一つのシミュレーション・プロジェクトにおいて同一種類の要素モデルを複数の配置でき，配置されたそれぞれの要素モデルが独自の内部変数や出力値を持つことができることがある．また，シミュレーションを途中で止めた場合に，シミュレーションの状態を保存しておき，シミュレーション中止時の状態からシミュレーションを再開することができるようにしている．そのため，演算の実行を担当するクラスだけではなく，異なる値を持つことができるようにオブジェクト・プログラミングにおけるインスタンスを生成するクラスやシミュレーション中止中のデータを保持し，ハードディスク上に保存することを担当するクラス，複数の要素モデル等から特定のものを識別するためのクラスから構成されている．これらの機能を担当するのは，それぞれ演算モデルクラス，演算ファクトリクラス，演算データクラス，および演算モデル定義クラスである．下記にクラスごとの機能と役割およびシミュレーション実行時の演算処理について述べる．

5-5-1-1　演算モデル定義クラス

演算モデル定義クラスは，要素モデルの種別や名称を定義するとともに，他の要素モデ

ル等と区別するための識別子が定義されている．識別子には，ファクトリ識別子，モデル識別子，およびパターン識別子がある．識別子とはCommonMPのプラットフォームが要素モデル等を指定するとき，他の要素モデル等と区別するために使う文字列である．ファクトリ識別子は，要素モデルを生成するときに用いるもので，CommonMPにインストールされている要素モデルの実行形式ファイル（DLL: Dynamic Link Library）の間でユニークな値を持たなければならない．CommonMPでは同一の実行形式ファイルの中に複数の要素モデルを定義することができるが，モデル識別子は，プラットフォームが同一の実行形式ファイルの中で要素モデルを特定するときに用いる．よって，モデル識別子は同一実行形式ファイルの中で定義されている要素モデルの個数分用意し，同一実行形式ファイルの中でユニークな値をとらなければならない．パターン識別子は，プラットフォームが同一要素モデルの中で伝送情報を特定するための文字列である．よって，パターン識別子は，同一要素モデルの中で定義されている伝送情報の種類分用意し，同一要素モデルの中でユニークな値をとる必要がある．

5-5-1-2　演算モデルファクトリクラス

演算モデルファクトリクラスは，後述する「演算データクラス」や「演算モデルクラス」のインスタンスを生成するとともに，当該モデルの持つパラメータやタイムステップ（演算時間間隔），伝送情報やモデルの概要等のモデル情報を設定し，表示する役割を持つ．ここで生成すべき演算データクラスや演算モデルクラス等を特定する際に，演算モデル定義クラスで定義した各種の識別子が用いられる．演算モデルファクトリクラスは，演算モデル定義クラスと対で作られる．

5-5-1-3　演算データクラス

演算データクラスは，当該要素モデルの演算中のデータを保持するクラスであり，要素モデル１つに対して１個作られる．CommonMPは，要素モデルの演算状態をファイルに保存し，一旦シミュレーションを中断しても，途中からシミュレーションを再開できるように設計されているが，そのためには演算データクラスにシミュレーションを再開するのに必要なデータをすべて記述しておかなければならない．

5-5-1-4　演算モデルクラス

演算モデルクラスは，演算を実施するクラスである．要素モデル１つに対して１個の演算モデルクラスが作られる．演算モデルクラスには，基底クラスとして「5-3-2　要素モデルの演算の進め方に関するタイプの設定」で述べた「現状計算型要素モデル」と「未来計算型要素モデル」の２種類が用意されており，要素モデル開発者はどちらかの基底クラスを継承して要素モデルを実装する．

演算モデルクラスには，パラメータ等の要素モデルのプロパティ情報を受け取るメソッド（SetProperty），接続情報（受信・送信それぞれ）をチェックするメソッド（ReceiveConnectionCheck，SendConnectionCheck），シミュレーションが始まる前に要素モデルを初期化するメソッド（Initialize），１演算時間間隔分の演算を実施するメソッド（Calculate），および送信情報を配信するメソッド（DataFusion）が定義されており，要素モデル開発者は，それぞれのメソッドをオーバーライド（上書き）することにより実装する．

5-5-2　要素モデルの演算処理

演算要素モデルの演算処理に関わるクラスのおおまかな役割と演算処理実行時の手順について述べる．演算処理の手順は構築時と演算実行時に分けて述べる．

5-5-2-1 演算処理の基本構造

CommonMP のシミュレーション実行時には，演算モデルクラスと演算データクラス，および要素間伝送データが主に関係し，それぞれプラットフォームの演算系の制御からの指示（メソッドコール）を受けて動く（図-5.12）．演算モデルクラスは１演算時間間隔の演算を実施するクラスであり，演算データクラスは演算実行中のデータを保持するクラス，要素間伝送データは，要素モデル間を伝送されるデータを表す．

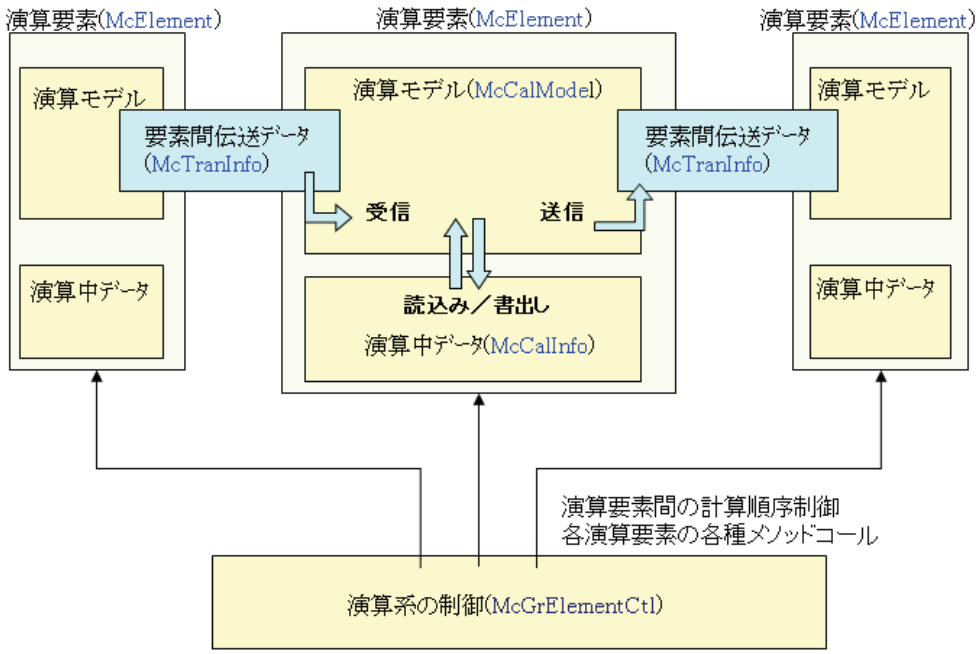


図-5.12　要素モデルの演算動作の基本構造

5-5-2-2　演算モデル構築時の処理

演算モデル構築時には，演算ファクトリクラスにより要素モデルや演算データおよび要素モデルの実体(インスタンス)，プロパティ情報，初期化情報等が生成される(図-5.13)．要素モデル構築の処理手順は以下のとおりである．①最初に演算モデルファクトリクラスにより要素モデルおよび演算データの実体（インスタンス）が生成されるとともに，要素モデルの初期化情報，プロパティ情報，およびモデル情報（要素モデル概要等）が作成される．②次に演算モデルクラスの SetCalInfo メソッドが呼び出され，演算データがセットされ演算モデルクラスから利用できるようになる．③要素モデルのプロパティ情報（パラメータ等)が演算モデルクラスの SetProperty メソッドにより要素モデルにセットされる．④受信情報の接続が演算モデルクラスの ReceiveConnectionCheck メソッドによりチェックされる．⑤同様に演算モデルクラスの SendConnectionCheck メソッドによりチェックされる．⑥演算モデルクラスの Initialize メソッドにより演算モデルの初期状態かセットされる．以上の手順により演算要素モデルが演算を実行する準備が整う．

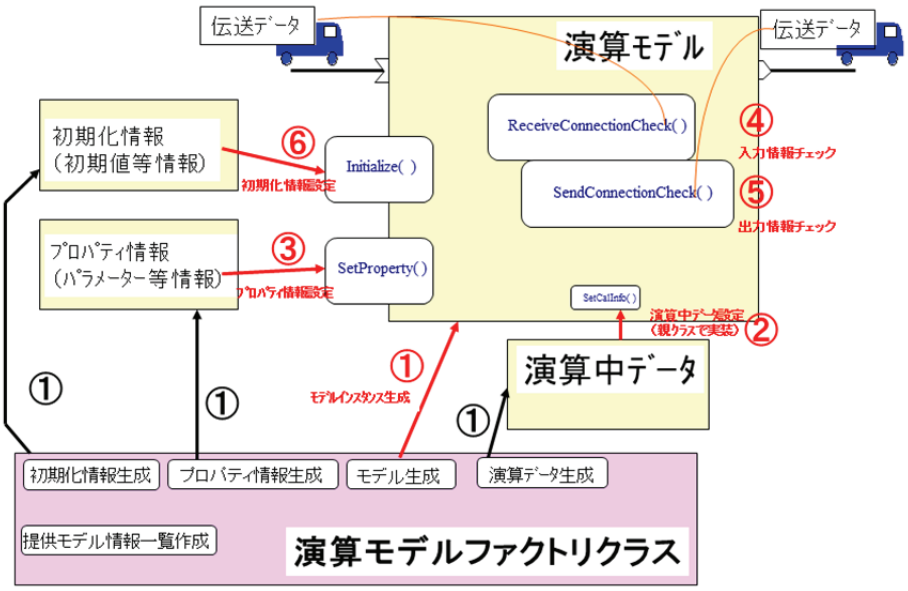


図-5.13　演算要素モデル構築時の処理手順

5-5-2-3　演算実行時の処理

演算要素モデルの演算実行時は,主に演算モデルクラスと演算データクラスが関与する．演算実行の処理手順は以下のとおりである（図-5.14)．①演算モデルクラスの ReadyCalculation メソッドが呼び出される．このメソッドは，シミュレーション実施上必須ではないが，要素モデル開発者が必要に応じて自由に実装することができる．②次に目標時刻に達したかを判断する IsConverged メソッドが呼び出される．目標時刻に達してい

た場合は演算を中止し，目標時刻に達していなかった場合は，演算処理を継続する．③演算モデルクラスのCalculateメソッドが呼び出され,1演算時間間隔分演算が実行される．④演算モデルクラスのDataFusionメソッドが呼び出され,演算結果を格納した伝送データが伝送路に保管される.⑤演算モデルクラスのChangeDeltaTimeAutomaticallyメソッドが呼び出され，必要に応じて演算時間間隔を変更する（実装は要素モデル開発者が任意に行うことができるが，通常は実装されない）．以下，②のIsConvergedメソッドにより演算が中止されるまで，②から④の手順を繰り返す．⑥演算を終了するときに，演算モデルクラスのCompleteCalculationメソッドが１度呼び出される．このメソッドの実装は，要素モデル開発者が必要に応じて行うことができる．

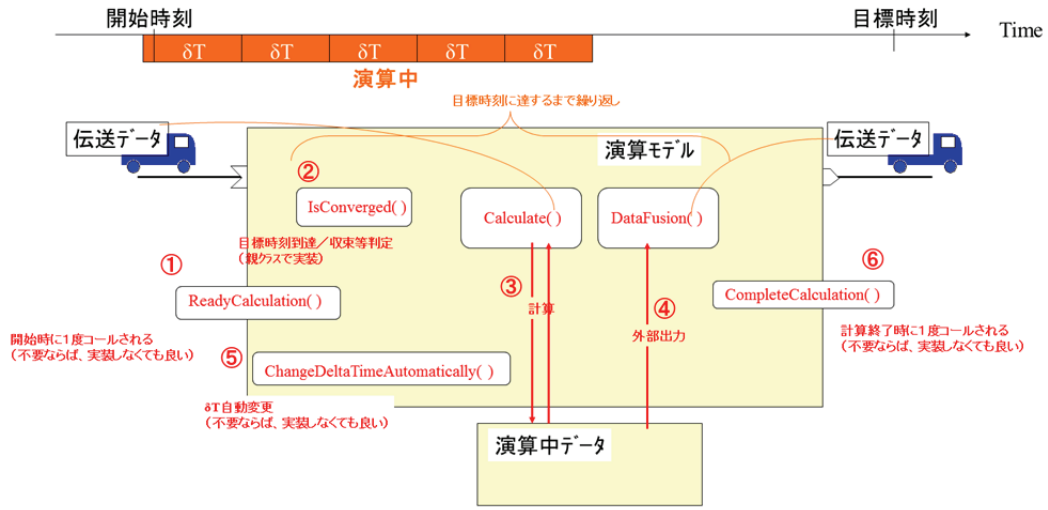


図-5.14　演算要素モデルの演算実行時の処理手順

5-6　要素モデルのラッピング

第4章で述べたように，CommonMPの機能要件として過去のプログラム資産を有効活用することが挙げられている．ここでは，水理・水文解析によく使われているFortranで記述されたプログラムをCommonMP上で制御する機能を設計することとする.一般に異なる仕様で記述されたプログラム同士を互いに連携させて動かすには，両者の間にインターフェースを挟んで,通信内容を翻訳する必要がある.異なる仕様で記述されたプログラムを包み，内部構造を隠蔽し，あたかも同じ仕様のプログラムのように見せる技術のことをラッピング（Wrapping）と呼び，ラッピングするプログラムのことをラッパーと呼ぶ．ここでは，CommonMPでFortranプログラムを通常のC#で記述された要素モデルと同じように動かすことができるFortranラッパーを設計することとする．

5-6-1　ラッピングの機能要件

ここでは，通常の要素モデルと同じように動かすことの機能要件として，以下の①から

⑤に定める機能要件を挙げた．簡易なラッピング手法既存では，①の演算時間間隔の設定ができないことがあり，また Fortran のような非オブジェクト指向のプログラミング言語では，同一のプログラムで複数の内部状態を持つように実装することが難しいので，機能要件として明示することにした．

ラッピング手法の主な機能要件

① Fortran プログラム側ではなくて，プラットフォーム側で演算時間間隔を設定できること．
② 同一種類の複数の要素モデルを配置できることとともに，それぞれの要素モデルは異なるパラメータや状態量を持つことができること．
③ CommonMP でサポートしている伝送情報の型をすべて取り扱うことができること．
④ シミュレーション・プロジェクト構築時において，Fortran のソースコードを開示する必要がないこと．
⑤ 特定の Fortran コンパイラに依存した手法ではないこと．

5-6-2　代表的なラッピング手法

代表的なラッピング手法として，データ受け渡し方式と動的リンク方式を挙げて，それらの特徴をまとめた（表-5.3）．双方の方式とも，演算を実行する部分（Fortran プログラム）とプラットフォームとの通信を仲介する部分（ラッピング要素モデル）からプログラムが構成される（図-5.17）．他の要素モデルとの通信もラッピング要素モデルを介して行われる．

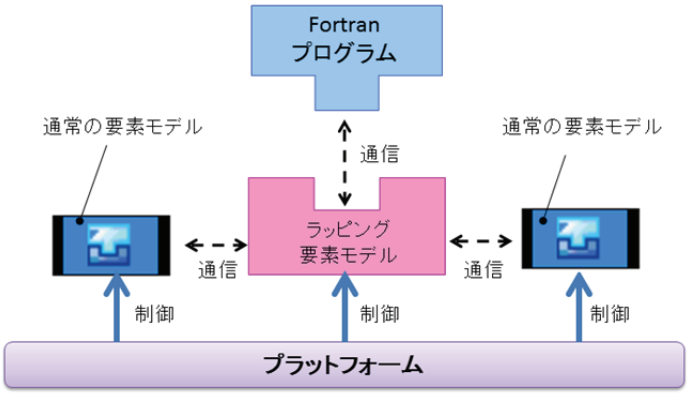


図-5.15　Fortran ラッパーの構造

データ受け渡し方式は，既存のプログラムをほぼそのまま使うもっとも簡易な手法である．通常の要素モデルはプラットフォームから制御を受けるが，この手法は Fortran プログラムを独立して動かせる方法である．演算に必要なデータをラッピング要素モデルを介

してプラットフォームから受け取るときは，プラットフォーム側と同期をとる必要があるが，この手法では演算に必要なデータが供給されるまでFortranプログラムが演算をストップさせて待つことにより，同期をとっている．既存のFortranプログラムの修正が少ないが，ラッピングされるプログラムの演算時間間隔をプラットフォームから設定できない等，制約が多い方法である．

動的リンク方式は,通常のCommonMPの要素モデルと同様にCommonMPから呼び出せるDLL（ダイナミックリンクライブラリ）を作成する方法である．この方法では，通常の要素モデルに近い動作が可能となり，演算速度も比較的速いという長所がある．しかしながら，ラッピング要素モデルを介してプラットフォームと通信できるように，プラットフォームのメソッドコールに対応したプログラム構造にする必要がありソースコードの修正量が多くなる．他のプログラミング言語で開発されたプログラムとの通信方法に関するコードの記述方法はFortran コンパイラによって異なり，Fortran プログラムの修正については，コンパイラ依存する．

表-5.3　代表的なラッピング手法の特徴

	データ受け渡し方式	動的リンク方式※
概　　要	プラットフォームからは完全に独立した Fortran プログラムを動かす方法であり，演算に必要なデータについては両者が互いに通信することで，送受信をする． Fortranプログラムからの通信方法により，共有メモリ方式や名前付きパイプ方式，ファイル交換方式等の方法がある．	Fortran プログラムにより，DLL（ダイナミックリンクライブラリ）を作成し，プラットフォームから DLL にアクセスすることにより，演算を実施する方法．
長　　所	Fortranコードの修正量が比較的少なくて済む．	通常の CommonMP の要素モデルと同じ方式であり，通常の要素モデルに近い動作が可能となる．演算速度は，比較的速い．
短　　所	プラットフォームと Fortran で演算時間間隔を揃える必要がある． 待ち合わせによるデータ連携方式なので，プラットフォームによる制御ができない． データ受け渡しの待ち時間のため，演算速度は遅くなる． 要素モデルが複雑な接続関係を持つと，設定が煩雑になる．	通常のプラットフォームのメソッドコールに対応したプログラム構造にする必要があり，ソースコードの修正量が多くなる． .NetFramework 上の他のプログラムとの通信方法に関する記述は Fortran コンパイラによって異なり，コンパイラ依存する方法である．

※ この他に静的リンク方式も考えられるが，この方法はプラットフォームのソースコードと要素モデルのソースコードを一つの実行形式ファイルにするものであり，CommonMP 上では実装できないので，ラッピング手法の範疇には入れないことにする．

以下に，代表的なラッピング手法のデータ受け渡し方式の中から名前付きパイプと動的リンク方式について，実際に Fortran プログラムをラッピングして性能（実行速度，ソースコードの修正量等）を検証した．

5-6-3　名前付きパイプ方式

名前付きパイプとは，複数のプロセス間でデータをやりとりするための共有メモリ領域のことで，キュー（先入先出：FIFO（First In First Out））型のデータ構造をしている．名前付きパイプは，ファイルに読み書きするように利用できるため，実装が容易であり，簡易なラッピングには適している．

5-6-3-1　名前付きパイプ方式の動作機構

名前付きパイプ方式では，ラッピング要素モデルと Fortran プログラムが名前付きパイプを介してデータファイルをやりとりする．名前付きパイプには１方向１種類のデータファイルを１個作成する必要がある．ラッピング要素モデルに対して双方向のデータのやりとりがある場合は，最低２個の名前付きパイプ（出力用と入力用）を作成する必要がある．

名前付きパイプ方式では，Fortran プログラムは CommonMP プラットフォームとは独立したプログラムであるので，シミュレーションを始める前に何らかの手段で Fortran プログラムを起動させておく必要がある．CommonMP プラットフォームには独立したプログラムを起動させる機能はないので，ラッピング要素モデルに Fortran プログラムを起動させる機能を実装した（図-5.16）．

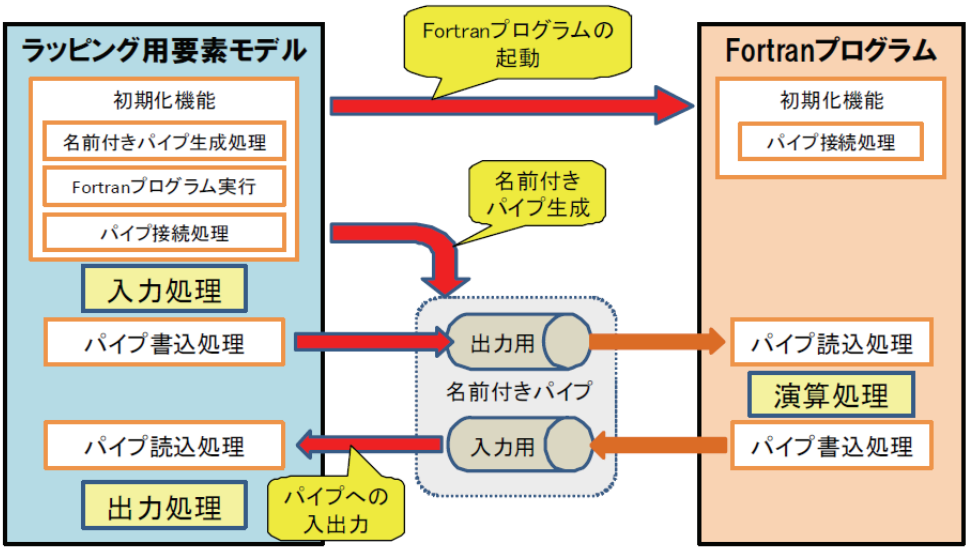


図-5.16　名前付きパイプ方式の動作機構 [20)]

ラッピング要素モデルと Fortran プログラムはそれぞれ独立したプログラムとして稼働するが，何らかの手段で時間同期をとる必要がある．CommonMP の演算要素モデルでは，1 演算時間間隔分の演算処理が実装されているが，一般的な水理・水文解析プログラムは時間ループを繰り返すように実装されている．名前付きパイプ方式では，Fortran プログラムに演算に必要なデータが保存されるまで演算を停止し，ラッピング要素モデルと Fortran プログラムの演算時間間隔（⊿t）をそろえることで，時間同期をとっている（図-5.17）．

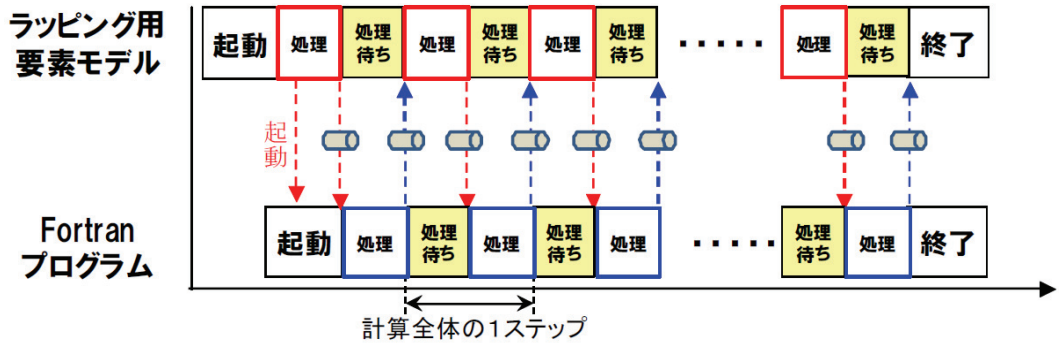


図-5.17　ラッピング要素モデルと Fortran プログラムの時間同期 [20)]

5-6-3-2　名前付きパイプ方式の実装

名前付きパイプ方式の実装環境を下記に示す（表-5.4）．C#コンパイラは Microsoft Visual Studio 2008 Professional Edition であり，有償のソフトウェアであるが，無償版の Express Edition を用いても同じことができる．Fortran コンパイラは無償のものを用いており，OS 以外は無償のもので再現することが可能である．ラッピングの対象とする Fortran プログラムは，土木学会水理公式集例題プログラム集河川編 2-2 ダイナミックウェーブの一次元不定流モデルである．

表-5.4　名前付きパイプ方式の実装環境

項　　目	仕　　様
プラットフォーム	CommonMP Ver1.2
OS	Windows 7 Professional 64bit
ミドルウェア	.NET Framework 4
C#コンパイラ	Visual Studio 2008 Professional
Fortran コンパイラ	GNU Fortran 4.7.1
Fortran プログラム	ダイナミックウェーブ式一次元不定流モデル

ラッピング要素モデルは，CommonMP プラットフォーム上で動く要素モデルである．ただし，通常の要素モデルと異なる点として，本来の演算処理は行わず，名前付きパイプを用いた Fortran プログラムと通信する機能をもつことである．ラッピング要素モデルには，通常の要素モデルとして機能の他に，下記の①から③の機能を実装する．

ラッピング要素モデルに実装する機能（追加分）

① 名前付きパイプを生成する機能
② 名前付きパイプにデータを入出力する機能
③ Fortran プログラムを起動させる機能

Fortran プログラムには，名前付きパイプへのデータの入出力の処理を実装する必要がある．名前付きパイプへのデータの入出力自体は，テキストファイルにデータを入出力するのと同じような要領で実装するとこができる．Fortran プログラムには，主に下記の①から③の３つの処理を追加する必要がある．

Fortran プログラムに追加する機能

① 名前付きパイプをオープンする機能
② 名前付きパイプからデータを入力する機能
③ 名前付きパイプへデータを出力する機能

通常の水理・水文解析プログラムは，初期設定した後，演算処理のための時間ループを回し，１演算時間間隔ずつ処理を進めるという方法をとっている．名前付きパイプ方式では，初期設定を行う場所にパイプをオープンする処理を実装し，時間ループの中で１演算時間間隔ごとにパイプから演算に必要なデータを入力し，パイプへ演算結果のデータを出力するという処理を実装すればよい（図-5.18）．

ダイナミックウェーブ一次元不定流モデルのソースコード（216 行）にラッピングのために追加したソースコードの量は 92 行（変数の宣言：5 行，パイプのオープン：30 行，ループ処理：10 行，パイプ入力：26 行，パイプ出力：10 行，パイプのクローズ：11 行）（画面出力用の記述およびコメント行を含む）であった．

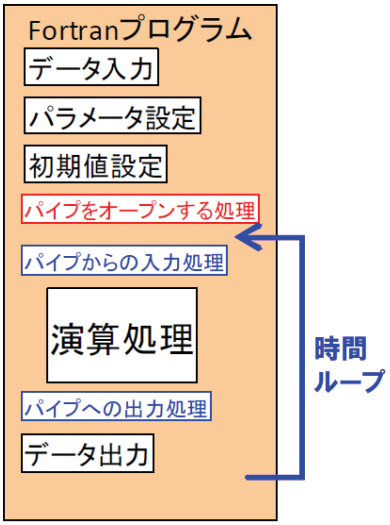


図-5.18　Fortran プログラムの修正 [20)]

5-6-3-3　名前付きパイプ方式の動作確認

名前付きパイプ方式によりラッピングしたダイナミックウェーブの Fortran プログラムを CommonMP プラットフォームから実行し，適正に動作していることが確認できた（図-5.19）.

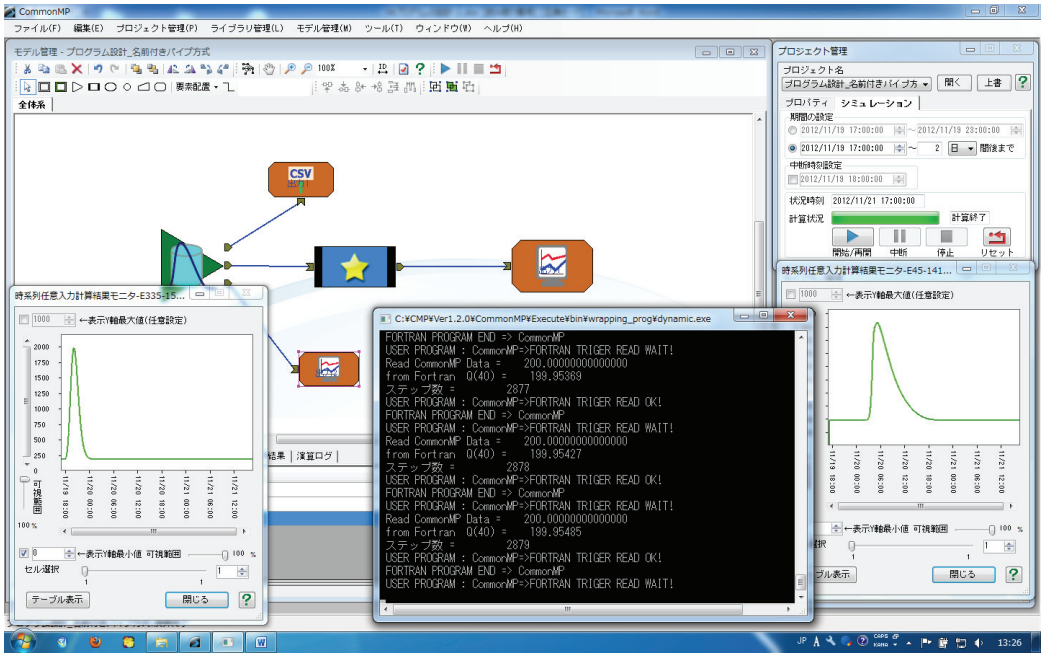


図-5.19　名前付きパイプ方式の動作確認

また，ラッピングをしていないFortranプログラムとラッピングをしたFortranプログラムの速度比較を実施した（表-5.5）．ラッピングされたFortranプログラムについては，演算速度の低下の原因となると考えられる画面出力のコードは削除して演算速度を比較することにした．比較に使用したFortranプログラム等の仕様および演算に要した時間は下記のとおりであった．オリジナルのFortranプログラムが上流端流量の入力データをプログラムの中で発生させているのに対して，ラッピングされたFortranプログラムは，上流端流量の入力データを他の要素モデルを生成したデータを伝送情報として受け取る方式なので，ラッピングされたFortranプログラムの方が不利な条件であり，一概には比較できないが，それを考慮しても名前付きパイプ方式のラッピングは，演算速度を遅い方法と言える．

表-5.5 演算性能等の比較

	オリジナル Fortran プログラム	ラッピングされた Fortran プログラム
演算時間※	0.0478 sec	12.1 sec
不定流モデルへの入力データの渡し方	同一プログラム内で生成した数値を変数で渡す.	他演算要素モデルが生成した数値を伝送情報として渡す.
モデル仕様	ダイナミックウェーブ式一次元不等流モデル＋上流端流量生成モデル	
断面形状	矩形	
断面数	51	
演算時間間隔	72 sec	
シミュレーション期間	２日	
CPU・計算機メモリ	Intel Core i7 (2.8GHz), 6.00GB	

※ オリジナルFortranプログラムは，CPU_TIME関数により計測，ラッピングされたFortranプログラムについては，手動計測3回の平均値

5-6-4 動的リンク方式

動的リンク（ダイナミックリンク）とは，プログラム実行時に初めて結合されるリンク方式のことで，動的リンク方式によって結合されるライブラリ（汎用性の高い複数のプログラムを再利用可能な形でひとまとまりにしたもの）をダイナミックリンクライブラリ（DLL）という．CommonMPの通常の要素モデルはDLLから構成されており，動的リンク方式によりラッピングされたFortranプログラムは,通常のCommonMPの要素モデルに近い動作が可能となる.一般的なFortranコンパイラには,DLLを作成する機能が備わっている.

5-6-4-1　動的リンク方式の動作機構

動的リンク方式では，CommonMP プラットフォームによりラッピング要素モデル（CommonMP 要素モデル部）が制御される各種処理内容に対して，ラッピング要素モデルから演算部分（初期化，データ取得，1 演算時間間隔分の演算処理等）を Fortran プログラムの DLL（FortranDLL 部）を呼び出すことにより実行する方法である（図-5.20）．オリジナルの CommonMP の要素モデルも DLL により構成されているので，通常の CommonMP の要素モデルに似た動きができるが，下記に挙げる制約事項があり，これらの制約を取り除くには設計上の工夫が必要である．

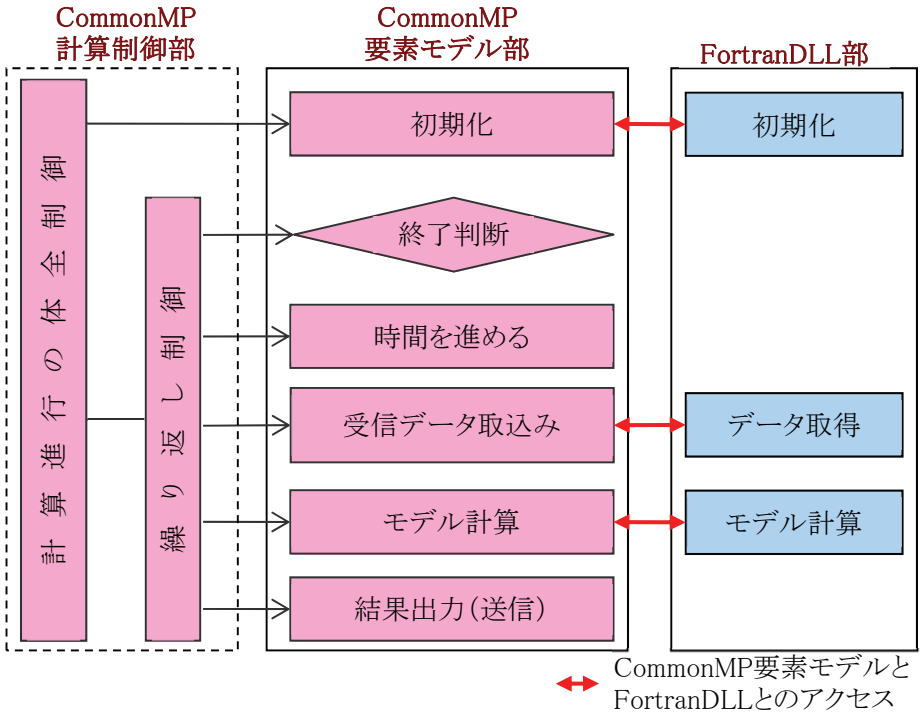


図-5.20　動的リンク方式の動作機構

動的リンク方式の制約事項

① Fortran プログラムを同一シミュレーション・プロジェクトで複数用いる場合でも，Fortran プログラムは複数のラッピング要素モデルに対して同じ内部変数しか持つことができない．
② シミュレーション中断時の状態を保存できず，シミュレーション中断状態からシミュレーションを再開することができない．
③ Fortran プログラム内での異常終了（例外）を CommonMP プラットフォーム側に伝え

ることができない.

①の制約については，Fortran のような手続き型プログラミング言語では，オブジェクト指向のプログラミング言語と異なり，１つのオブジェクトに対して，複数のインスタンスを持つことができないことが原因である．しかしながら，シミュレーション・プロジェクト上で同一要素モデルを複数利用することができることが，CommonMP の機能要件として本質的であるので，この制約については解消することを検討する．②の制約については，CommonMP の機能要件としては本質的でなく，もともとオリジナルの Fortran プログラムにも入っていない機能であるので，検討対象とはしなかった．③についても CommonMP の機能要件としては本質的ではないし，異常終了の発生の情報はプラットフォームを経由しなくても，Fortran プログラムが直接コンソール画面に出力することも可能であるので，検討対象からは外した．

5-6-4-2　動的リンク方式の実装

動的リンク方式の実装環境を下記に示す（表-5.6）．C#コンパイラは Microsoft Visual Studio 2008 Professional Edition であり，有償のソフトウェアであるが，無償版の Express Edition を用いても同じことができる．Fortran コンパイラも有償のものを用いているが，基本的に DLL を出力できる Fortran コンパイラならば同じことができ，無償の GNU Fortran でも可能である．これらの環境は，OS 以外は無償のもので再現することが可能である．ラッピングの対象とする Fortran プログラムは，独自に開発した貯留関数法流出モデルである．

表-5.6　動的リンク方式の実装環境

項　　目	仕　　様
プラットフォーム	CommonMP Ver1.2
OS	Windows 7 Professional 32bit
ミドルウェア	.NET Framework 4
C#コンパイラ	Visual Studio 2010 Professional
Fortran コンパイラ	Visual Fortran 64 Composer(32bit)
Fortran プログラム	貯留関数法流出モデル

Fortran ラッピングをする際に，もっとも大きい制約は 5-6-4-1①の Fortran プログラムがそれぞれのラッピング要素モデルに対して個別の内部変数を持てない点である．その制約については，オリジナルの Fortran プログラムの内部変数をラッピング要素モデルのメンバー変数として持つとして持ち，ラッピング要素モデルから Fortran プログラムのサブ

ルーチンの引数として１演算時間間隔ごとに受け渡しすることである（図-5.25）．このようにすれば，異なるラッピング要素モデルからFortranプログラム（DLL）が呼ばれても，それぞれのラッピング要素モデルに対してそれぞれの内部状態量を持つプログラムとして値を返すことができる．

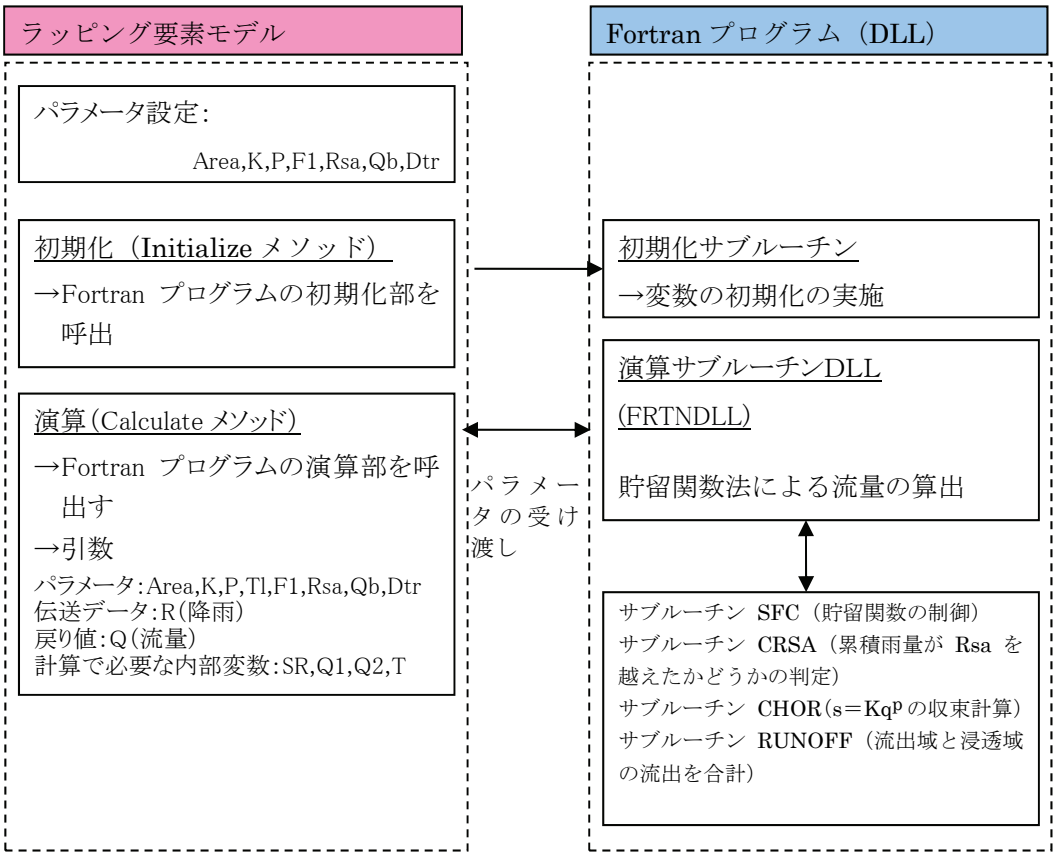


図-5.25　動的リンク方式の実装内容（貯留関数法流出モデルの例）

ラッピング要素モデルの実装においては，通常のCommonMPの要素モデルの実装の他に，ラッピングの対象となるFortranプログラムのDLLを指定するとともに，演算モデルクラスのInitializeメソッドにおいて，Fortranプログラムの初期化サブルーチン呼び出すための記述を追加することと，Calculationメソッドにおいて Fortranプログラムの演算を担当するサブルーチンを呼び出し，演算に必要な水理・水文量のデータやパラメータを引数として渡すための記述を追加することが必要である．

水理・水文量のデータやパラメータを格納するための変数やDLLのファイル名は，個々のFortranプログラムごとに異なるため，この方法では個々のFortranプログラムごとに専用のラッピング要素モデルを作成する必要がある．それでは，ラッピング要素モデルの汎用性が確保できないので，汎用性を確保するためには，ラッピング要素モデルに対して

対象とする Fortran プログラムの DLL 名を定めておくとともに，Fortran プログラムの初期化部と演算部のラッピング要素モデルから呼び出されるサブルーチン名や引数も指定しておく，引数は使う，使わないにかかわらず多めに確保しておくようにするなどの対応が考えられる．

通常の水理・水文解析プログラムは，通常，初期化した後，演算に必要な時系列データの取り込み，モデルによる演算処理，演算結果の出力のサブルーチンで構成される時間ループを回す構造になっている．動的リンク方式の実装に用いる貯留関数法流出モデルの Fortran のオリジナルのプログラムもこのような構造になっているので，ラッピング要素モデルからの呼び出しに対応できるようにサブルーチンを再構成する必要がある（図-5.24）．Fortran オリジナルプログラムの修正は，下記に示した手順に従う．

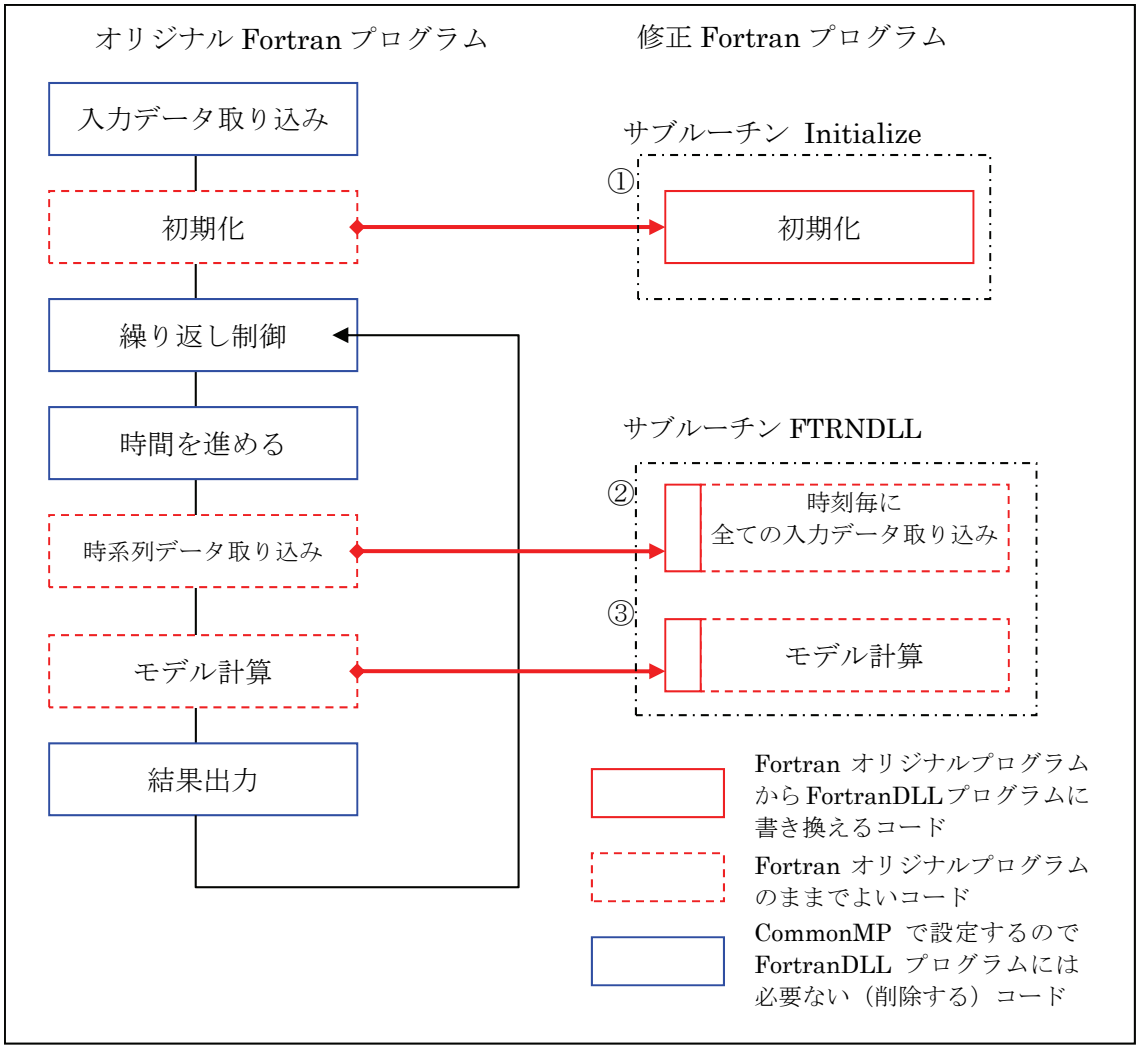


図-5.24　オリジナル Fortran プログラムと修正された Fortran プログラムの構造比較

動的リンク方式による Fortran プログラムの修正手順

① 初期化実行部の作成

内部変数の初期化の宣言を記述する.

② 入力・出力データの伝達部の作成

サブルーチン FRTNDLL の引数に，ラッピング要素モデルと演算に必要な変数群をすべて記述する.

③ モデル演算部の切り出し

オリジナル Fortran プログラムの時間ループを抽出し，演算時間間隔ごとの演算処理をサブルーチン FRTNDLL に記述する.

貯留関数法流出モデルのオリジナル Fortran プログラムのソースコード（204 行）に対して，ラッピングのために初期化や演算処理のために抽出して作成したサブルーチンのコードの量は 152 行であった．時間ループの処理等が不要になったため，ソースコードの量は減少しているが，プログラム構造が大幅に変化しているのでソースコードの修正量は多い.

5-6-4-3　動的リンク方式の動作確認

動的リンク方式によりラッピングした貯留関数法の Fortran プログラムを CommonMP プラットフォームから実行し，適正に動作していることが確認できた（図-5.27）.

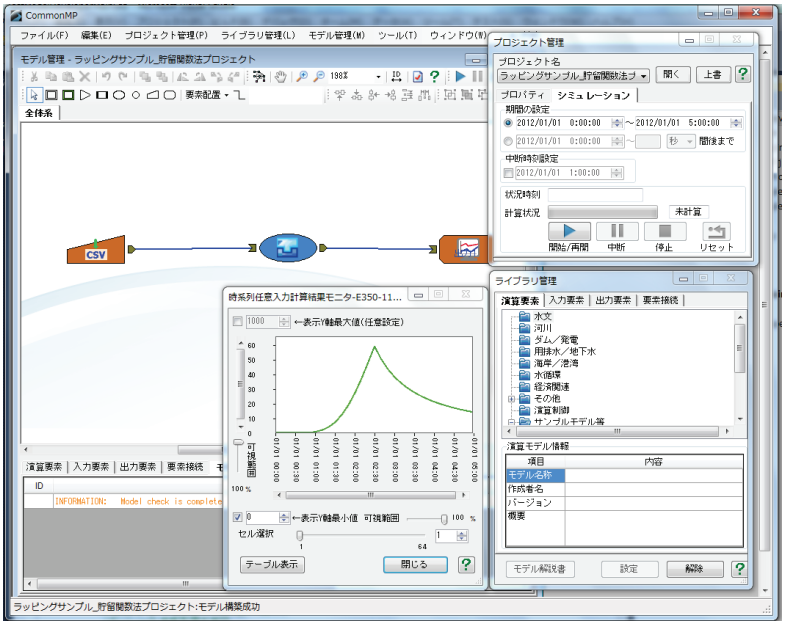


図-5.27　動的リンク方式の動作確認

また，オリジナルの Fortran プログラムとラッピングをした Fortran プログラムの速度比較を実施した（表-5.7）．比較に使用した Fortran プログラム等の仕様および演算に要した時間は下記のとおりである．オリジナルの Fortran プログラムは同一プログラム内で生成した数値を変数で渡しているのに対して，ラッピングされた Fortran プログラムは，雨量データをテキストファイルから入力要素モデルを介して伝送情報として受け取るようにしている．データの読込み方式についてはラッピングされた Fortran プログラムに不利な条件であり，一概に比較できないが，ラッピングすることにより演算速度が低下しているのが確認された．名前付きパイプ方式よりは，演算速度の低下は著しくないと言える．

表-5.7 演算性能等の比較

	オリジナル Fortran プログラム	ラッピングされた Fortran プログラム
演算時間※	1.56E-02 sec	1.3 sec
入力データの読込み	同一プログラム内で生成した数値を変数で渡す.	テキストファイルから
モデル仕様	貯留関数法式流出モデル	
演算時間間隔	600 sec	
シミュレーション期間	5 時間	
CPU・計算機メモリ	Intel Core i5-3470(2.80GHz), 4.00GB	

※ オリジナル Fortran プログラムは，CPU_TIME 関数により計測，ラッピングされた Fortran プログラムについては，手動計測 3 回の平均値

5-7 機能拡張ツール

機能拡張ツールとは，CommonMP プログラムがシミュレーション実行のために制御する要素モデルとは別に，独立したプログラムとして CommonMP 上で稼働させることのできるソフトウェアのことを言う（図-5.28）．具体的には，シミュレーション実施前または実施中に外部のデータベースからデータを取得し，入力要素モデルに提供したり，シミュレーション終了後にシミュレーション結果を可視化し，表示したりするソフトウェアのことである．CommonMP のツールメニューから呼び出されるハイドロ／ハイエト表示ツールや構造定義ファイル編集ツールが機能拡張ツールに相当する．CommonMP-GIS も機能拡張ツールに分類できる．機能拡張ツールは，CommonMP に予めインストールされているものもあるが，要素モデルと同様にユーザが独自に開発して，CommonMP にインストールすることもできる．こ

こでは，機能拡張ツールの一つとして，水文水質データ取得ツール[21]をとり上げ，その設計について述べる．

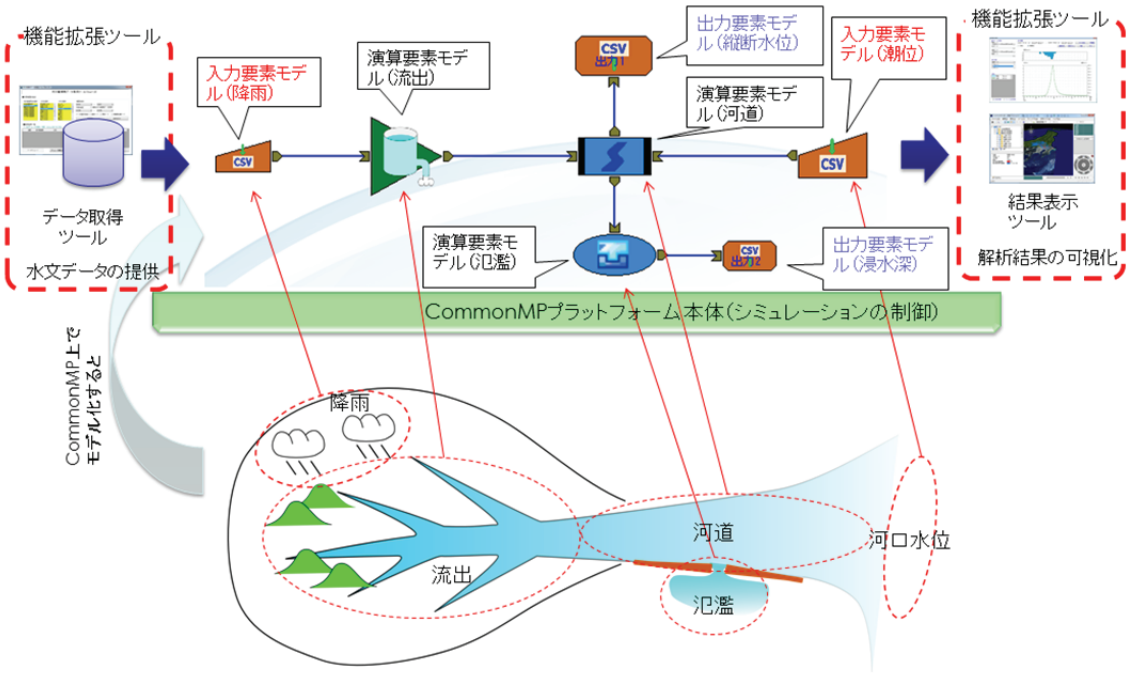


図-5.28　機能拡張ツール

5-7-1　水文水質データ取得ツール

水文水質データ取得ツールとは，水文水質データベース[8]に格納されているデータを取得し，ローカルのパーソナルコンピュータ（PC）に保存するためのツールである．水文水質データベースはインターネット上に公開されており，一般ユーザはウェブブラウザを用いて閲覧したり，水文水質データをダウンロードしたりすることができる．しかしながら，水文水質データベースからのウェブブラウザによるデータの取得は，データベースの管理上の理由により一括して取得できるデータの量等を制限しており，長期間に渡るデータが必要となる統計解析への利用や水理・水文解析ソフトウェアからの利用には適していなかった．また，水文水質データベースには過去データの他に，10 分ごとに更新されるリアルタイムデータが格納されており，ウェブブラウザによりダウンロードすることが可能である．リアルタイムデータについては，洪水浸水予測等のため，アプリケーションソフトウェアから利用するニーズが高いので，水文水質データベースには河川 GIS・河川アプリケーション標準インターフェースガイドラインに基づくインターフェース（以下，「標準インターフェース」と呼ぶ）が整備されており，これに対応したアプリケーションソフトウェアからの問い合わせによりデータを一括してダウンロードすることが可能である．水文水

質データ取得ツールは，CommonMP を水文水質データベースに整備された標準インターフェースに対応させるための機能拡張ツールである（図-5.29）.

図-5.29　水文水質データ取得ツールのユーザインターフェース

5-7-1-1　セキュリティ対策

水文水質データベースは，水文観測業務規程[9]に基づくデータの公開という重要な機能を持っているため，一般へのリアルタイムデータや過去データを提供する場合，データの提供処理によって水文水質データベースに負荷がかかり，システムが不安定になることを防ぐ必要がある。また，インターネットからデータを取得することを考えると，悪意あるユーザからの不正アクセス防止等セキュリティを考慮した提供方法にすることが重要である。

そこで，水文水質データベースのミラーサーバを構築し，リアルタイムデータや過去データの提供をミラーサーバから行うこととした．なお，ミラーリングは，水文水質データベースのスナップショット機能(任意の時間におけるデータベースのデータを取得する機能)を利用し，1 日に 1 回程度の頻度で行うこととした．リアルタイムデータについては，10 分ごとに統一河川情報システム[22]から配信されるので，ミラーサーバにも直接配信し，保存されるようにした．

悪意あるユーザからのアクセスを防ぐため，ミラーサーバへのアクセスに際してユーザ認証を施し，登録されたユーザのみがアクセスできるようにした（図-5.30）.

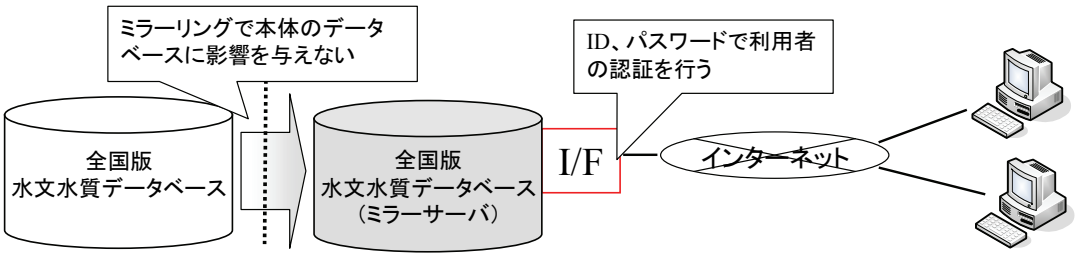


図-5.30　水文水質データベースのセキュリティの確保

5-7-1-2　リアルタイムデータの取得・保存

統一河川情報システムからは，テレメータデータは 10 分間隔で提供されており，水文水質データベースのリアルタイムデータもその提供間隔に合せて更新されている．これに対し，水文水質データ取得ツールは，1 分間隔で水文水質データベースを参照するようにしている.データベースを参照する頻度を高くすると,その分データベースに負荷がかかり，参照頻度を低くすると，データベースから取得したリアルタイムデータのデータが生成された時刻とクライアント PC にリアルタイムデータが保存されたときの時刻のずれが大きくなる.両者のバランスを勘案し,水文水質データベースへの参照時間間隔を 1 分とした.

リアルタイムデータは，洪水予測等のために使われるため，CommonMP を長期間に渡って稼働させることが想定される．リアルタイム処理中にリアルタイムデータをクライアント PC に保存し続けると，やがてはハードディスクの容量がいっぱいになってしまい，ComomnMP を安定させて稼働させることは難しい．一度利用したデータは一定期間経過したらハードディスク上から削除する必要がある．水文水質データ取得ツールで取得されるリアルタイムデータは 1 観測所ごとにファイルを分けて保存される．新たに取得されたリアルタイムデータはファイルの末尾に追記され，6 個以上（10 分更新なので，1 時間以上）前のデータはファイルから削除される．このようにして，リアルタイムで保存されたファイルが無尽蔵に大きくなることを防いでいる．

5-7-1-3　データ取得の制限の設定

ユーザに与えるデータのダウンロードの制限（データの期間や観測所数）が少ないほどデータ取得ツールとしてのユーザビリティがよくなる一方で，データベースサーバへの負荷が高くなる．ここでは，ユーザの利便性とデータベースサーバの負担を勘案し，ユーザに与えるデータのダウンロードの制限を検討した．検討に当たっては，水文水質データベースの本サーバを用いて接続負荷試験を行った（表-5.8）.

表-5.8 接続負荷試験の結果

		水文水質データ取得ツール		処理時間	
		期間	観測所数	同時接続 10 の場合	同時接続 100 の場合
リアルタイムデータ		10 分	50	14 秒	2 分 20 秒
過去	10 分データ	7 日間	50	9 秒	2 分 6 秒
	時間データ	1 年間	50	2 分 57 秒	32 分 32 秒
	日データ	10 年間	50	54 秒	9 分 49 秒
	年統計データ	全期間	50	31 秒	5 分

接続負荷試験の結果を踏まえ，一括して取得できるデータの制限を下記のとおり設定した (表-5.9)．参考までにウェブブラウザによるデータのダウンロードの際の制限を併記する．ウェブブラウザを用いたデータのダウンロードに比べてデータの取得期間と観測所数において大量のデータを一括してダウンロードすることが可能になっている．

表-5.9 データの一括ダウンロードの制限

		水文水質データ取得ツール		ウェブブラウザ	
		期間	観測所数	期間	観測所数
リアルタイムデータ		－	50	－	1
過去	10 分データ	7 日間	50	7 日間	1
	時間データ	1 年間	50	31 日間	1
	日データ	10 年間	50	1 年間	1
	年統計データ	全期間	50	全期間	1

5-8 マルチプロジェクト演算

マルチプロジェクト演算とは，複数のシミュレーション・プロジェクトを同時並行に実行させる演算方式である．洪水予測システムの運用では，リアルタイムデータを取得し，初期値を変えたシミュレーションを5分おきや10分おきの定間隔に実行させることがある．シミュレーション期間が長い場合やシミュレーションにおいて大量の演算を行わなければならない場合は，次のシミュレーションを開始するまでに前に実行させていたシミュレーションが完了しているとは限らず，複数のシミュレーションを重複させて実行させる必要

が出てくる（図-5.31）.

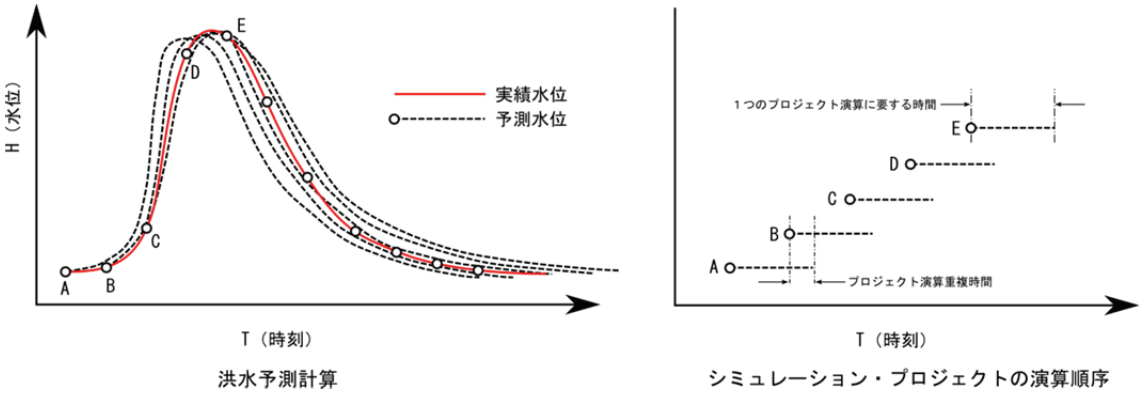


図-5.31　マルチプロジェクト演算

通常の利用においても，シミュレーション・プロジェクトの実行時間が長い場合，バックグラウンド処理ができないと，シミュレーション・プロジェクトを実行中は他のシミュレーション・プロジェクトの編集作業ができず，ユーザビリティが悪くなる．この場合においても，マルチプロジェクト演算ができるとバックグラウンド処理が自動的に可能となり，ユーザビリティの改善が可能である．

ここでは，マルチプロジェクト演算の演算方式について検討するとともに，複数のプロジェクトの開始や終了を制御する方法やプロジェクト間のフィードバック等の機能の設計を行い，これらの機能確認のために，卓上簡易予測システムを作成し，運転試験を行った．マルチプロジェクト演算の性能を評価するため，負荷試験も行った．

5-8-1　演算方式

マルチプロジェクト演算の演算方式としては，マルチスレッド方式とマルチプロセス方式がある（図-5.32）．ここでは，それぞれ方式について長所・短所を整理した上で，CommonMP プラットフォームに実装する方式について検討する．

マルチスレッド方式とは，１つのアプリケーションソフトウェア上で複数のスレッド（CPU の処理単位）を同時並行に稼動させる方式である図-5.32 左）．CommonMP でマルチスレッド方式を実行する場合は，複数のシミュレーション・プロジェクト（これらがスレッドに当たる）を１つの CommonMP プラットフォーム上で同時並行に稼動させることを意味する．

マルチプロセス方式は，１つの PC 上に複数のプロセス（アプリケーションソフトウェアと言ってもよい）を同時並行に稼動させる方式である．CommonMP を用いてマルチプロセス方式を実行する場合は，１つの PC 上で複数の CommonMP を起動させ，それぞれの CommonMP 上で１つのシミュレーション・プロジェクトを実行させる（図-5.32 右）．

図-5.32　マルチスレッド方式とマルチプロセス方式

マルチスレッド方式は，異なるスレッド（シミュレーション・プロジェクト）同士で計算機メモリ共有することができるので,プロジェクト間の通信の実装が容易である.また，同一アプリケーションソフトウェア上で複数のシミュレーション・プロジェクトを稼動させることができるので，アプリケーションソフトウェアからシミュレーション・プロジェクトを制御しやすい利点がある．その一方で，シミュレーション・プロジェクト間の独立性が低いので，アプリケーションソフトウェアとしての安定性はシングルスレッドのアプリケーションと比較すると低くなる．マルチスレッド方式は，CommonMP プラットフォームの GUI をそのまま用いることができるので，高い操作性を求められる GUI を用いたデスクトップ・アプリケーションに適した方法である．

マルチプロセス方式は，シミュレーション・プロジェクト間で計算機メモリを共有することはできないが，シミュレーション・プロジェクトの独立性が高く，システムとしての安定性が高い方法である．複数のシミュレーション・プロジェクトの制御は，単一のアプリケーションソフトウェアから行うことはできず，プラットフォームを跨いだ制御が必要になるので，マルチスレッド方式に比べると実装は難しい．マルチプロセス方式は，高い安定性が求められるサーバ・アプリケーションに適した方法である．マルチスレッド方式とマルチプロセス方式の長所・短所等を整理すると，表-5.10 のようになる．

CommmonMP は，当初からデスクトップ・アプリケーションとして開発が進められてきた経緯から，ユーザが机上で洪水イベントごとに要素モデルのパラメータ等を調整するような使い方を想定し，マルチプロジェクト演算の方式としてマルチスレッド方式を採用することとした．しかしながら，操作性よりも安定性が重視される洪水予測の基幹システム等に CommonMP を導入するためには,マルチプロセス方式によるマルチプロジェクト演算の実現が必要である．

表-5.10 マルチスレッド方式とマルチプロセス方式の長所・短所等

	マルチスレッド方式	マルチプロセス方式
操作性	高い	低い
安定性	低い	高い
主な用途	デスクトップ・アプリケーション	サーバ・アプリケーション

5-8-3 マルチプロジェクト演算制御の設計

マルチプロジェクト演算ができるようになると，複数のプロジェクトの開始や終了を制御する方法やプロジェクト間のフィードバック等の機能を設計する必要が出てくる．ここでは，これらのシミュレーション・プロジェクトの制御やフィードバック処理等のプロジェクト間の通信方式を設計した．

マルチスレッド方式によるマルチプロジェクト演算は,一つの CommonMP プラットフォーム上で複数のシミュレーション・プロジェクトを稼動させる方式なので，マルチプロセス方式とは異なり，CommonMP アプリケーションソフトウェアの内部にマルチプロジェクト演算の制御部を設けることができる．

マルチスレッド化されたシミュレーション・プロジェクト群を制御する機能を TGC（Task Group Control）と呼ぶこととする．TGC の設計においては，既存の洪水予測の計算手法に対応できるように汎用性を確保する必要がある．洪水予測のためには，一定間隔で予測計算を行うことと，観測データを用いてフィードバックを行い，予測計算に反映させることが必要である．汎用性を確保しつつ，この機能を実現するため，TGC が持つべき機能として，①時間管理機能，②プロジェクト管理機能，③フィードバック処理機能の３つ定義した（表-5.11）．①の時間管理機能は，TGC の対象となっているシミュレーション・プロジェクト群を一定時間間隔で一定時間実行させるための機能である．洪水予測システムは，10 分ごとあるいは 30 分ごとのように一定時間間隔ごとに例えば 6 時間先までの予測計算を行うような運用をするため，この機能を設けている．②のプロジェクト管理機能は，TGC の管理対象となっているシミュレーション・プロジェクト群を構成する個々のシミュレーション・プロジェクトの演算実行順序やフィードバック情報の受け渡しに関する制御を行う．洪水予測システムでは，例えば過去から現在までの観測データを用いてパラメータを変えた複数のシミュレーションを実行させてパラメータを最適化させ，そのパラメータを用いて未来における予測計算を行うことがある．この場合，過去から現在まで複数のシミュレーション・プロジェクトを実行した後，最適化されたパラメータ（フィードバック情

報）を受け渡し，未来の予測計算用のシミュレーション・プロジェクトを実行する．プロジェクト管理機能は，これらのシミュレーション・プロジェクトの実行順序を制御するとともに，フィードバック情報の受け渡し（どのシミュレーション・プロジェクトからどのシミュレーション・プロジェクトへ受け渡すか）に関する制御を行う．③のフィードバック処理機能は，過去データによるパラメータの最適化等のフィードバック情報を作成する機能である．①の時間管理機能と②のプロジェクト管理機能は，CommonMP プラットフォーム側で用意した．②のプロジェクト管理機能については，シミュレーション・プロジェクトの実行順序やフィードバック情報の受け渡しは洪水予測の方法により異なり，一律に定めることが難しいので，ユーザをソースコードレベルでカスタマイズできるようにした．③のフィードバック処理機能は，プラットフォーム側で機能を持つのではなくて，シミュレーション・プロジェクトにより実現することとした．

表-5.11　マルチプロジェクト演算制御の機能

機能項目	機能概要	実現方法
時間管理機能	TGC に登録されたシミュレーション・プロジェクト群を一定時間間隔で実行し，終了させる機能．	プラットフォーム側で用意する．
プロジェクト管理機能	TGC に登録されたシミュレーション・プロジェクト群の中の個々のシミュレーション・プロジェクトの実施順次を設定し，制御する機能． フィードバック情報の受け渡しに関する設定および制御する機能．	プラットフォーム側で用意する．ただし，ソースコードレベルでユーザがカスタマイズできるようにする．
フィードバック処理機能	フィードバック情報を算出する機能．	ユーザがシミュレーション・プロジェクトとして作成する．

5-8-3　卓上簡易洪水予測システムの作成および運用試験

CommonMP に実装したマルチスレッド方式によるマルチプロジェクト演算機能およびプロジェクト間通信機能の機能確認のために，卓上簡易予測システムを作成し，運用試験を行った．

卓上簡易洪水予測システムとして，下記に示したフレームワークの構築を行った（図-5.33）．洪水予測の演算手順は，以下のとおりである．

① 予測に用いる水文データは，5-7-1 で示した水文水質データ取得ツールを用いて 10 分更新のリアルタイムデータを取得する．取得した水文データは，Wisef 形式でローカルの PC のハードディスクに保存する．

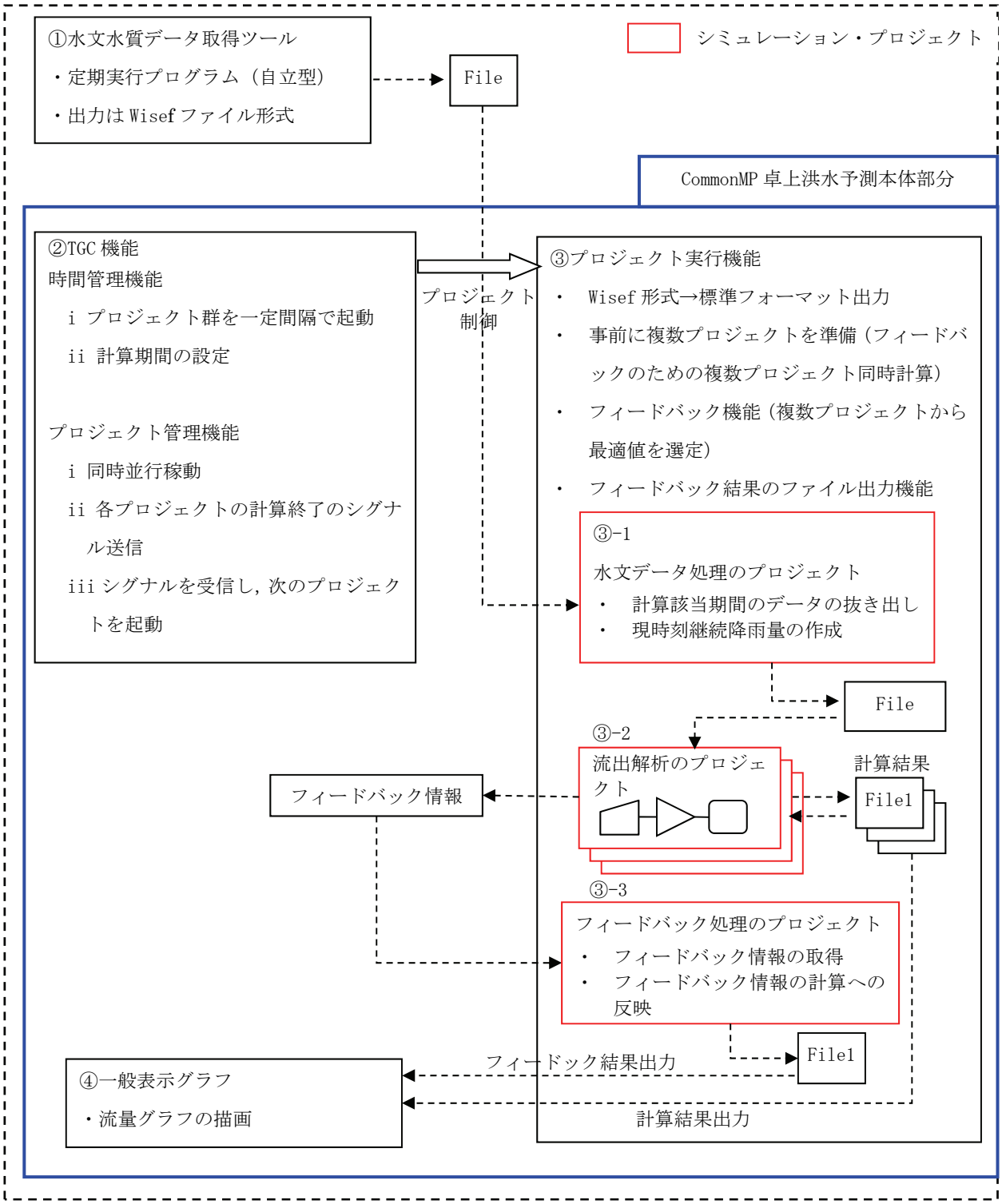


図-5.33　卓上簡易洪水予測システムのフレームワーク

② TGC により，シミュレーション・プロジェクト群を一定間隔で起動する．予測計算のための③に示すシミュレーション・プロジェクト群の中の個々のシミュレーション・プロジェクトを実行制御する．

③ TGC の制御により，シミュレーション・プロジェクト群の中の個々のシミュレーション・プロジェクトを実行する．

③-1 水文データ処理のシミュレーション・プロジェクトにより，ハードディスクに保存された水文データから演算に必要な期間のデータを切り出し，流域ごとの現在時刻における継続降雨量データを作成する．

③-2 パラメータを変えた複数の洪水予測のためのシミュレーション・プロジェクトを実行する．

③-3 水位（流量）観測データと③-2 で算出された流量データを比較することにより，フィードバック処理用のプロジェクトでフィードバック情報を作成する．フィードバック情報を③-2 のシミュレーション・プロジェクトに反映させ，未来の予測計算を行う．

④ 未来の予測計算が終了したら，結果表示のグラフを描画するとともに，②のステップに戻る（水文水質データ取得ツールは，TGC とは独立して稼動するプログラムであるので，②のステップには戻らない．）．

卓上簡易洪水予測システムの仕様および外観を下記に示す（表-5.12，図-5.34）．洪水予測の対象流域は東北地方の名取川とし，河道モデル 3 個，流出モデル 5 個，ダムモデル 1 個の簡易なモデル化を行い，流域シミュレーション・プロジェクトを作成した（図-5.35）．降雨データ処理のためのプロジェクト，流域シミュレーション・プロジェクトの流出率を 5 ケース設定した流域シミュレーション・プロジェクト，フィードバック処理のためのプロジェクト 2 個の全部で 8 個のプロジェクトから構成される．

降雨データ処理のためのプロジェクトは，水文水質データ取得ツールで取得され，Wisef ファイルとして保存された降雨量データを読み取り，流域シミュレーション・プロジェクト用の 5 つの流出モデルごとに継続時間雨量データを作成する（図-5.36）．

フィードバック処理のためのプロジェクトは，2 個作成した（図-5.35）．1 個目はフィードバック情報作成のためのプロジェクトである．このプロジェクトは，流出率を 0.4 から 0.8 までの 5 ケース設定した流域シミュレーション・プロジェクトの算出した 5 種類の名取橋地点における流量と水文水質データベースにより取得した同地点の水位を H-Q 関係式より変換した流量から式-5.1 により流域シミュレーション・プロジェクトごとの重み（ウェイト）を算出する．もう一つは，フィードバック情報を予測モデル（流域シミュレーション・プロジェクト）に反映させるためのプロジェクトである．このプロジェクトは，5

個の流域シミュレーション・プロジェクトの計算結果にフィードバック情報として算出した重みをかけ，予測計算値の最適値を算出する．

表-5.12　卓上簡易洪水予測システムの仕様

項　　目	内　　容
対象流域	名取川（河道モデル：3 個，流出モデル：5 個，ダムモデル：1 個）
演算プロジェクト	・雨量データ処理プロジェクト ・流域シミュレーション・プロジェクト（流出率を 0.4～0.8 まで変えた 5 ケース） ・フィードバック処理プロジェクト（フィードバック情報算出用，フィードバック情報反映用）
使用データ	水文水質データ取得ツールにより取得された Wisef 形式のリアルタイムデータ ・雨量データ：作並，二口，笹谷，上菅生，郡山 ・水位データ：名取橋
予測期間	・10 分間隔で 6 時間先まで予測（フィードバック処理のため 6 時間前から現在の値も計算）
結果表示	・6 時間前から 6 時間先まで，10 分間隔 ・10 分間隔で計算結果のグラフを更新

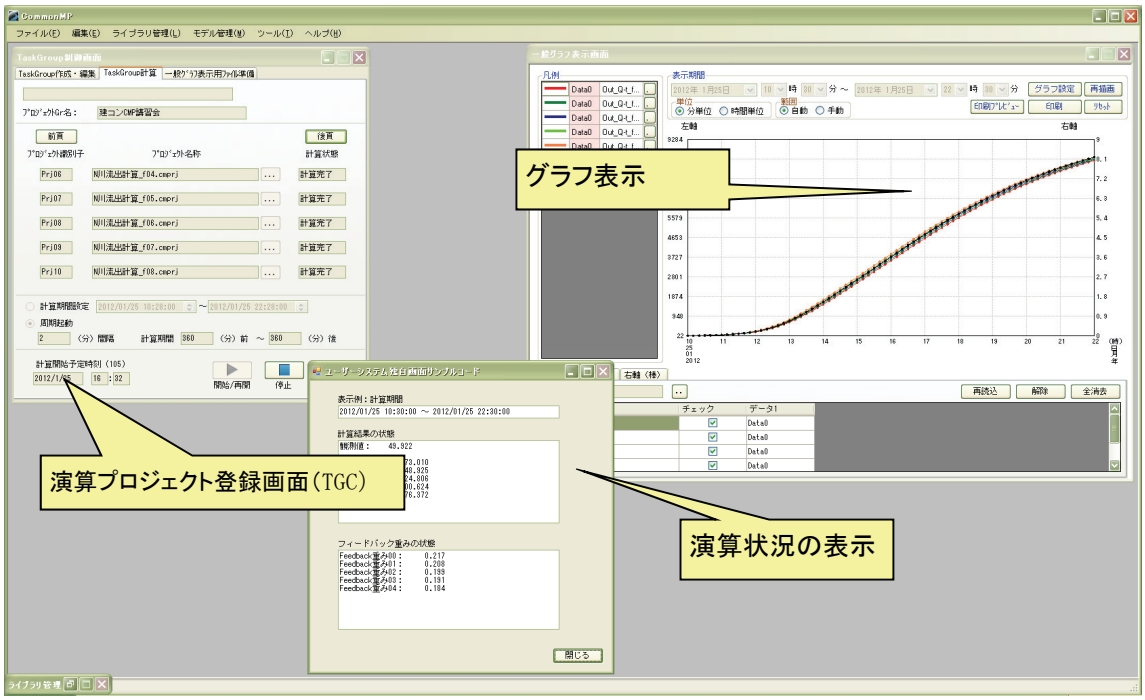


図-5.34　卓上簡易洪水予測システムの外観

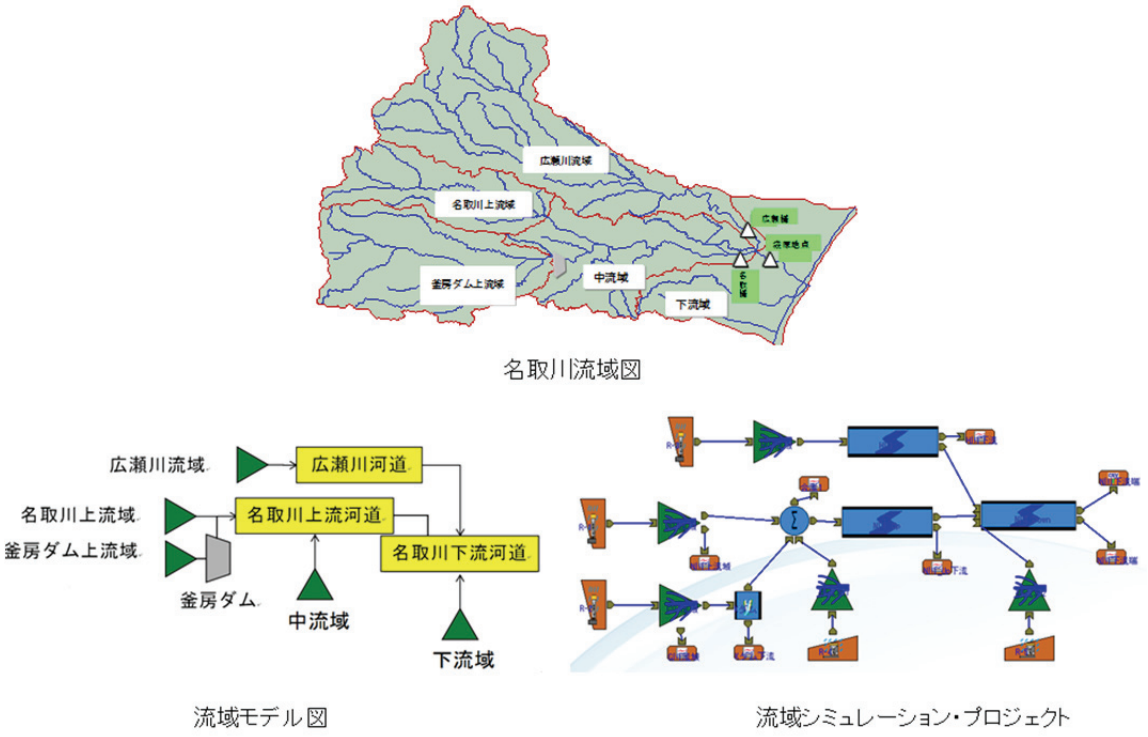


図-5.35　対象流域のシミュレーション・プロジェクト

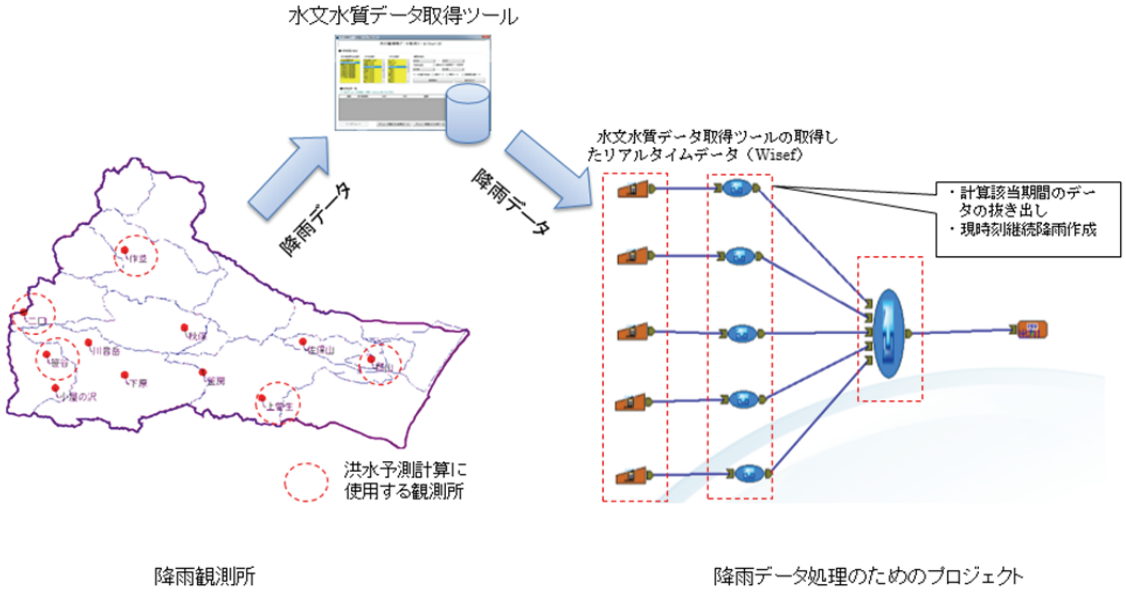
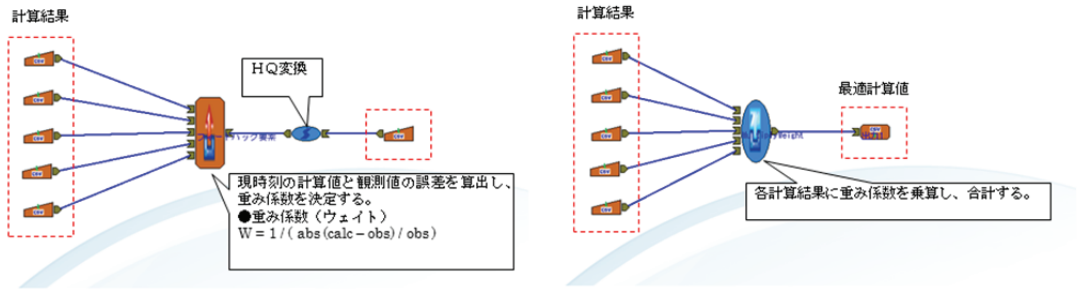


図-5.36　降雨データ処理



フィードバック情報を作成するプロジェクト　　フィードバック情報を反映させるプロジェクト

図-5.37　フィードバック処理プロジェクト

$$W = \frac{1}{|Q_{calc} - Q_{obs}| / Q_{obs}} \quad \cdots\cdots\text{式-5.1}$$

W: 重み（ウェイト）

Q_{calc}：流量予測値

Q_{obs}：流量観測値（H-Q 関係式により観測水位から変換）

式-5.1 により，計算値と観測値の差が小さい流域シミュレーション・プロジェクトほど，重みの値が大きくなる．

作成した卓上簡易洪水予測システムの運用上の安定性を確認するため，3 日間の連続運転を行った．運用試験に用いた PC の仕様は，通常の河川事務所の環境を想定し，下記のとおりとした（表-5.13）．

表-5.13　運用試験に用いた PC の仕様

項　目	仕　様
システム情報	OS ： Windows XP Professional Service Pack 3 [5.1 Build 2600] Memory ： 2043 MB
CPU	CPU Name ： Intel Celeron (Conroe-L) Name String ： Genuine Intel(R) CPU 575 @ 2.00GHz Platform ： LGA775 Number(Logical) ： 1 Feature ： MMX SSE SSE2 SSE3 SSSE3 XD Intel 64 Clock ： 1993.39 MHz L1 I-Cache ： 32 KB L1 D-Cache ： 32 KB L2 Cache ： 1024 KB [Full:1993.39 MHz]
HDD	IDE 320.0GB ST9320421ASG

3 日間の連続運用実施時に行った動作確認結果を示す．各機能ともに，動作確認項目において，問題なく動作することを確認できた（表-5.14）．

表-5.14　簡易卓上洪水予測システムの動作確認結果

機能項目	動作確認の内容	確認結果
水文水質データ取得	CommonMP ツールとしての稼動状況	○
	出力される Wisef データのフォーマット形式	○
	毎 10 分のリアルタイムデータの出力	○
	出力される Wisef データの入力要素モデルでの読込	○
予測計算	リアルタイムデータを使用した計算での結果出力	○
フィードバック情報作成	計算結果およびリアルタイムデータの取得	○
	重み計算とフィードバック情報の登録	○
フィードバック情報反映	フィードバック情報の取得	○
	他要素モデルへの重み係数の反映	○
グラフ表示	最新計算結果の更新・表示	○

○：問題なし

また，システムソフトウェアとしての安定性を検証するため，連続運用実施時に CPU，計算機メモリ、HDD（ハードディスク）の状態のシステムログの取得を行い，それらのシステムログ（図-5.38 から図-5.43）を下記の項目により評価した（表-5.15）．下記の評価のとおり，TGC 機能および演算プロジェクトともに安定して動作していることが確認できた．

・Processer Time（CPU 稼働率）

連続運用期間の最大でも 15%程度であり，TGC 機能および演算プロジェクトモデルともに大きな負荷が生じていないことが確認できた（図-5.38）．

・Interrupts/sec（ハードウェアの割り込み回数）

連続運用期間の平均として概ね 100 回/sec 程度であり，TGC 機能および演算プロジェクトともに大きな負荷が生じていないことが確認できた（図-5.39）．

※負荷が大きくなると数千万回/sec という単位でハードウェアの割り込みが発生する．

・Available Memory（利用可能な物理メモリ）

連続運用期間中若干減少はしているが，メモリリークに起因するような問題は発生しておらず，TGC 機能および演算プロジェクトともに安定して動作していることが確認できた（図-5.40）．

・Pages/sec（ハードページフォルトを解決するための HDD とのデータ授受）

連続運用期間中の平均として概ね数回程度であり，ハードページフォルトもほとんど見られず，TGC 機能および演算プロジェクトともに大きな負荷が生じていないことが確認できた（図-5.41）.

※負荷が大きくなると数百万回/sec という単位でハードページフォルトが発生する.

・Disk Read Time（読み込み要求でディスクがビジー状態だった割合）

連続運用期間中の平均として概ね数%程度であり、TGC 機能および演算プロジェクトともに大きな負荷が生じていないことが確認できた（図-5.42）.

・Disk Write Time（書き込み要求でディスクがビジー状態だった割合）

連続運用期間中の平均として概ね数%程度であり，TGC 機能および演算プロジェクトともに大きな負荷が生じていないことが確認できた（図-5.43）.

2.8 日付近で，各ログの負荷が大きくなっているっているが，定期的に見られる現象ではないため，洪水予測システムの連続稼動による負荷ではないと考えられる．当該負荷の原因としては,連続稼動を実施した PC のバックグラウンドで実行されているウイルスソフト等の定期実行が実施されたと考えられる．また，負荷自体も大きなものではない.

表-5.15　システムソフトウェアとしての安定性評価のために用いた指標※

項　　目		内　　容
CPU	Processor Time（%） （75% 未満が適正）	プロセッサがアイドル以外のスレッドを実行する時間のパーセンテージを表し，プロセッサの利用状況を示す主なインジケータとして使用できる.
	Interrupts/sec	プロセッサに対して発生しプロセッサが処理するハードウェア割り込みの秒あたりの平均数.
メモリ	Available Memory （Mbyte）	コンピュータ上で実行するプロセスが使用可能な物理メモリの容量を、メガバイト単位で表す.
	Pages/sec （0 に近いほど適正）	ハード ページフォルトを解決するためにディスクとの間で読み書きされるページの数を表す（ハード ページフォルトは、プロセスにおいて必要なコードまたはデータが作業セットまたは物理メモリの他の場所にも存在せず，ディスクから取得しなければならないときに発生する．）．このカウンタは，システム全体の遅延を招くページフォルトを示す主要なインジケータとして設計された. このカウンタの値が高いほど，ハード ページフォルト（OS がメモリを確保するためにディスクにアクセスする）がディスク I/O および CPU リソースに負担を与えていることを示す.
HDD	Disk Read Time(%)	読み込み要求の処理でディスクがビジー状態だった経過時間の割合
	Disk Write Time(%)	書き込み要求の処理でディスクがビジー状態だった経過時間の割合。

※Microsoft TechNet（http://technet.microsoft.com/）を編集

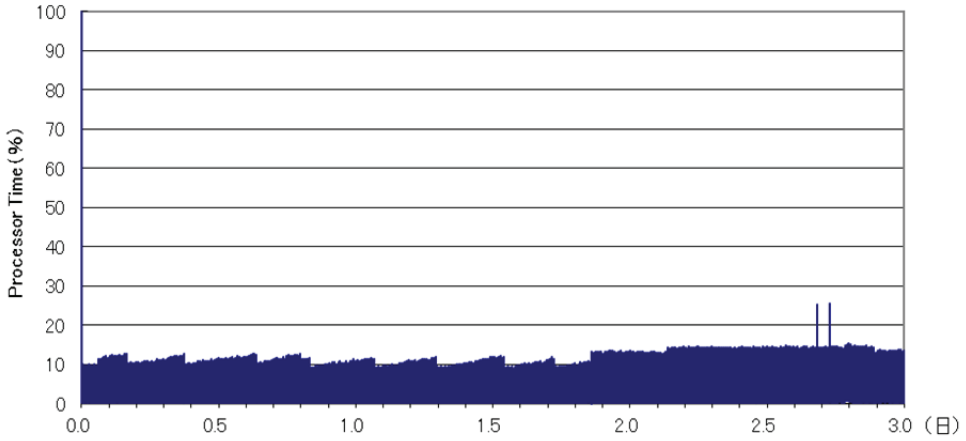


図-5.38　CPU 稼働率（Processor Time）

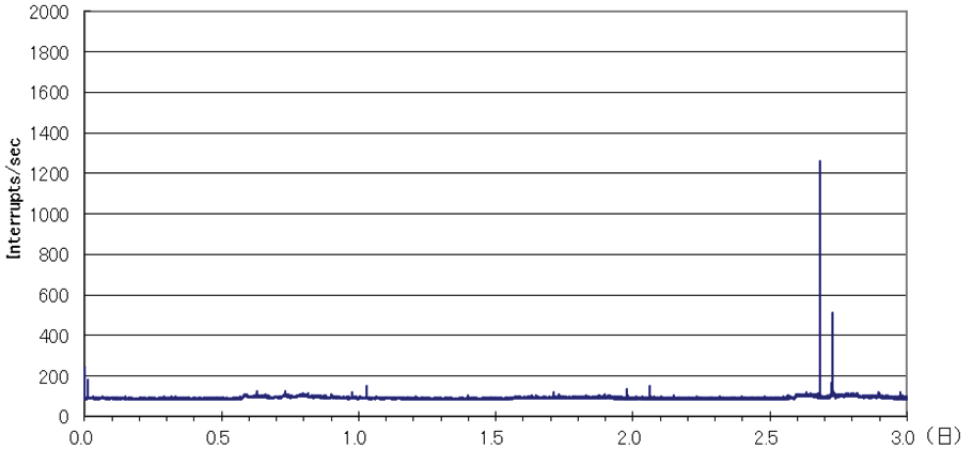


図-5.39　ハードウェアの割り込み回数（Interrupts/sec）

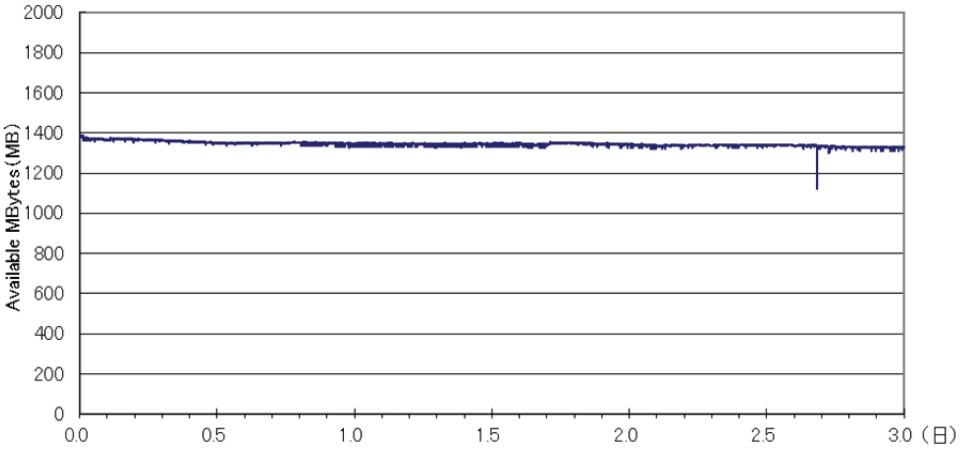


図-5.40　利用可能な物理メモリ（Available Memory）

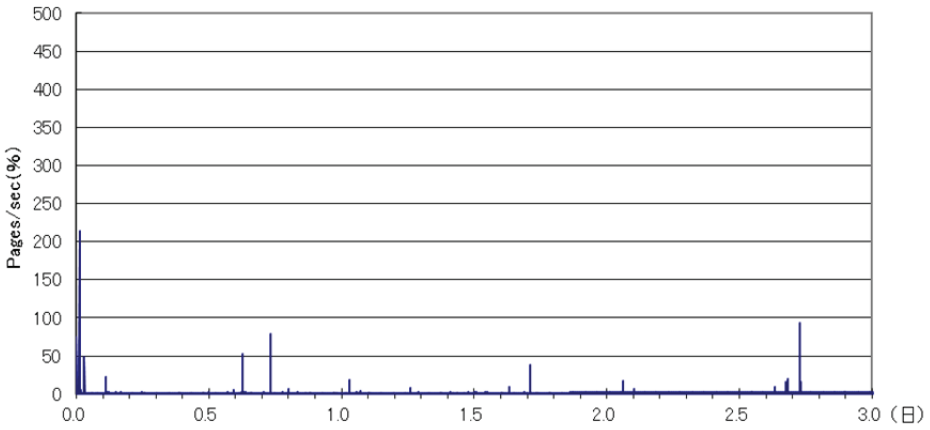


図-5.41　ハードページフォルトを解決するための HDD とのデータ授受（Pages/sec）

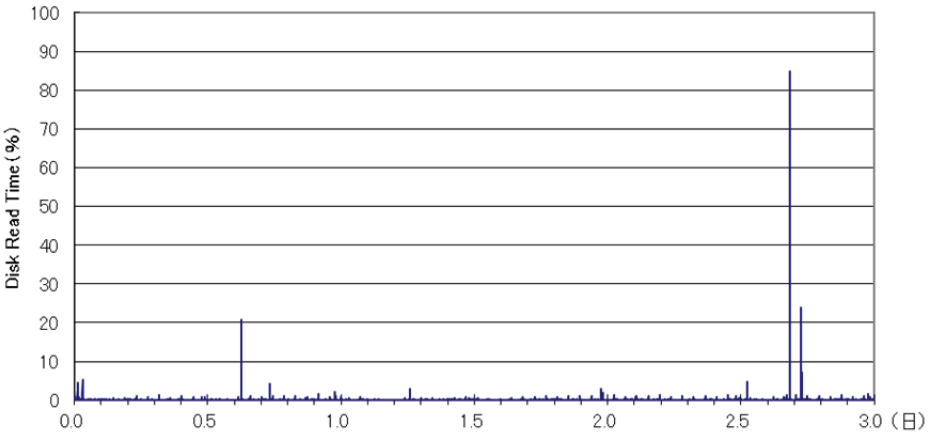


図-5.42　読み込み要求でディスクがビジー状態だった割合（Disk Read Time）

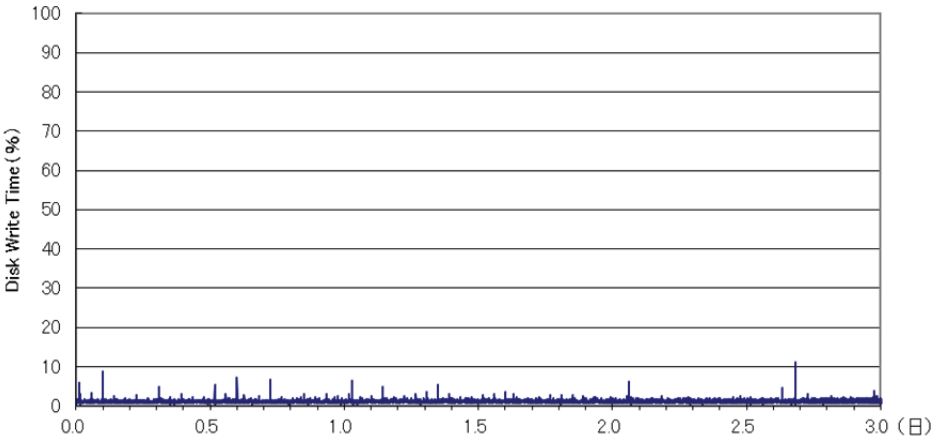


図-5.43　書き込み要求でディスクがビジー状態だった割合（Disk Write Time）

5-8-4　マルチプロジェクト演算の負荷試験

マルチスレッド方式によるマルチプロジェクト演算の性能を評価するため，5-8-3 の卓上簡易洪水予測システムのために作成した名取川の流域シミュレーション・プロジェクト（図-5.35）を用いて，複数の流域シミュレーション・プロジェクトを同時並行に実行させることによる負荷試験を行った。試験ケースとして，同時並行に実行させるシミュレーション・プロジェクトの数を 5，10，15 として 3 ケース実施した。下記にマルチプロジェクト演算の負荷試験の仕様を記す（表-5.16）.

表-5.16　マルチプロジェクト演算の負荷試験の仕様

項　　目	内　　容
使用したシミュレーション・プロジェクト	名取川洪水予測プロジェクト （河道モデル：3 個，流出モデル：5 個，ダムモデル：1 個）
マルチプロジェクト演算の実施方法	3 分間隔で起動，1 時間実施
同時並行に実施させるシミュレーション・プロジェクトの数	5，10，15 の 3 ケース

マルチプロジェクト演算性能の評価方法は，5-8-3 の卓上簡易洪水予測システムと同じにした（表-5.15）．実施結果を図-5.44 から図-5.61 に示す.

これら実施結果によると，特に大きな負荷が生じていないことが確認でき，本負荷試験で用いている程度のシミュレーション・プロジェクトの規模においては，パフォーマンスが著しく低下するなどの問題もなく演算が実行可能であることが確認された．また，同時並行に実施させるシミュレーション・プロジェクトの数の増加に比べて，PC への負荷が極端に増えるということもなかった.

・Processer Time（CPU 稼働率）

同時並行に実施させるシミュレーション・プロジェクトの数が多くなるにつれて CPU 稼働率も増えているが，シミュレーション・プロジェクトの数を 15 にした場合においても最大で 25%程度であり、大きな負荷が生じていないことが確認できた（図-5.44 から図-5.46）.

・Interrupts/sec（ハードウェアの割り込み回数）

同時並行に稼動させるシミュレーション・プロジェクトの数が多くなるにつれて若干

ハードウェアの割り込み回数も増える傾向にあるが，大きな負荷が生じていないことが確認できた（図-5.47 から図-5.49）.

※負荷が大きくなると数千万回/sec という単位でハードエェアの割り込みが発生する.

・Available Memory（利用可能な物理メモリ）

複数シミュレーション・プロジェクトを実行することで極端にメモリを確保してしまうことやメモリリークに起因するような問題も発生しておらず，安定して動作していることが確認できた（図-5.50 から図-5.52）.

・Pages/sec（ハードページフォルトを解決するための HDD とのデータ授受）

同時並行に実施させるシミュレーション・プロジェクトの数が多くなるにつれて若干ハードページフォルトが増加する傾向にあるが，大きな負荷が生じていないことが確認できた（図-5.53 から図-5.55）.

※負荷が大きくなると数百万回/sec という単位でハードページフォルトが発生する.

・Disk Read Time（読み込み要求でディスクがビジー状態だった割合）

同時並行に実施させるシミュレーション・プロジェクトの数が多くなるにつれて若干値が増加する傾向にあるが，大きな負荷が生じていないことが確認できた（図-5.56 から図-5.58）.

・Disk Write Time（書き込み要求でディスクがビジー状態だった割合）

同時並行に実施させるシミュレーション・プロジェクトの数が多くなるにつれて若干値が増加する傾向にあるが，大きな負荷が生じていないことが確認できた（図-5.59 から図-5.61）.

※各システムログに負荷が大きくなっている箇所が部分的に見られるが，定期的に見られる現象ではないため，洪水予測システムの連続稼動による負荷ではないと考えられる。

当該負荷の原因としては,連続稼動を実施した PC のバックグラウンドで実行されているウイルスソフトやアプリケーションの自動更新等の定期実行が実施されたと考えられる．また、負荷自体は大きなものではない.

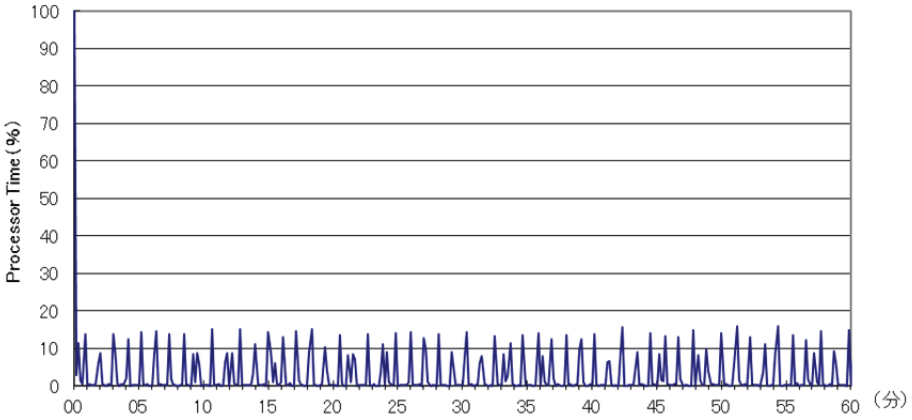


図-5.44　プロジェクト数 5

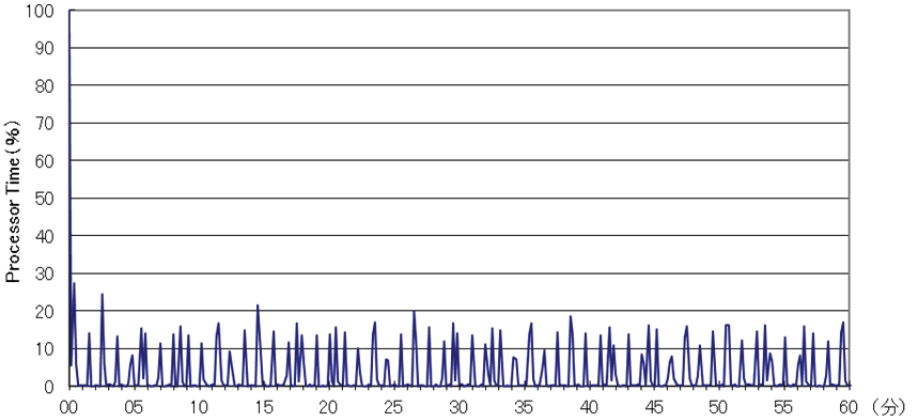


図-5.45　プロジェクト数 10

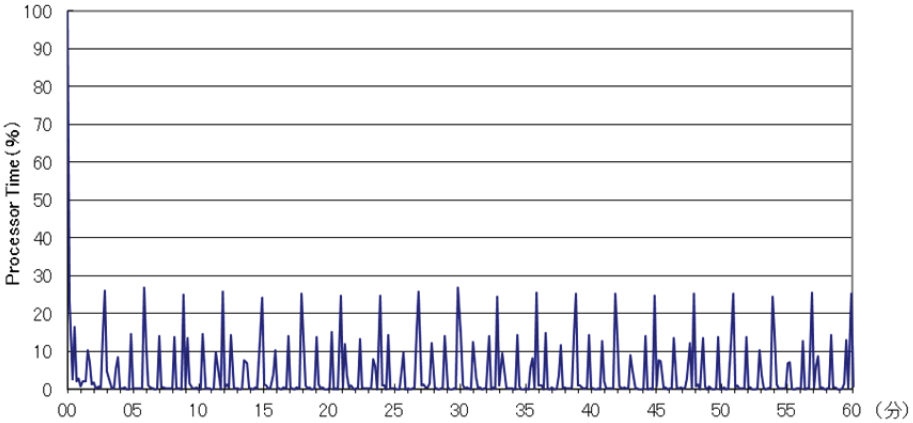


図-5.46　プロジェクト数 15

CPU 稼動率（Processor Time）の負荷試験結果（図-5.44 から図-5.46）

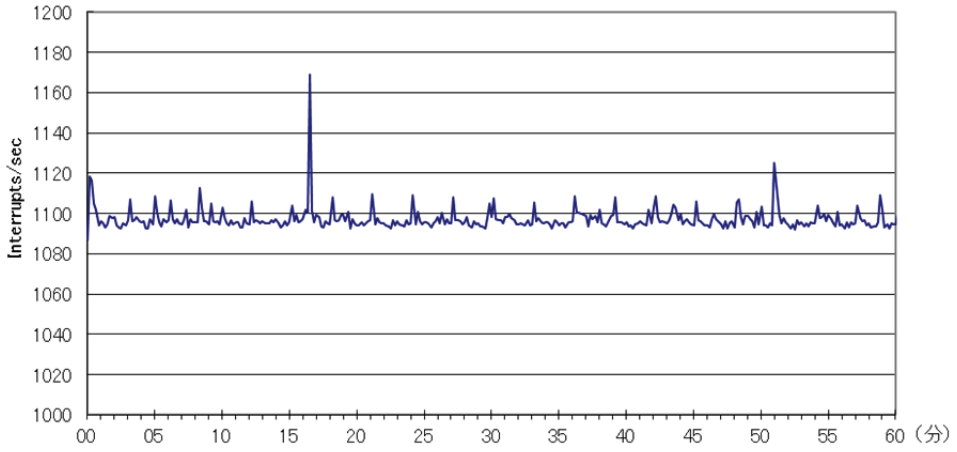


図-5.47　プロジェクト数 5

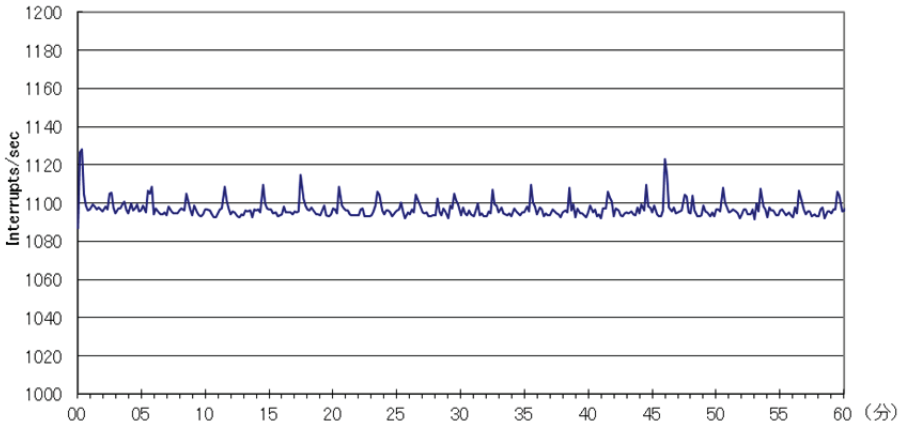


図-5.48　プロジェクト数 10

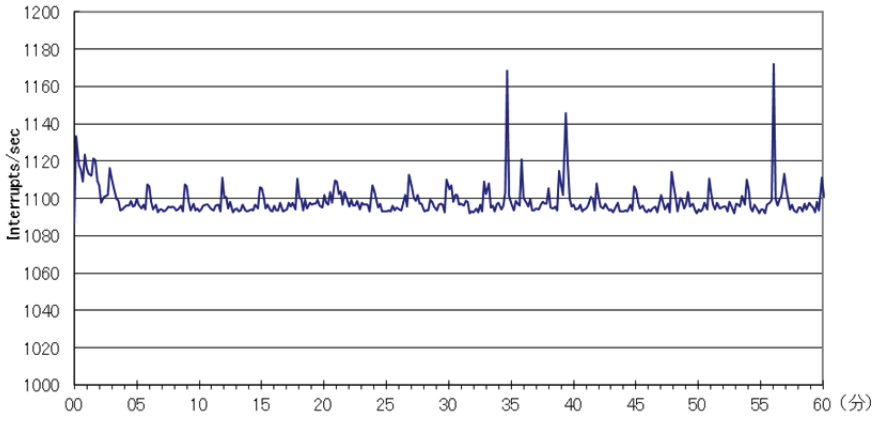


図-5.49　プロジェクト数 15

ハードウェアの割り込み回数（Interrupts/sec）の負荷試験結果（図-5.47 から図-5.49）

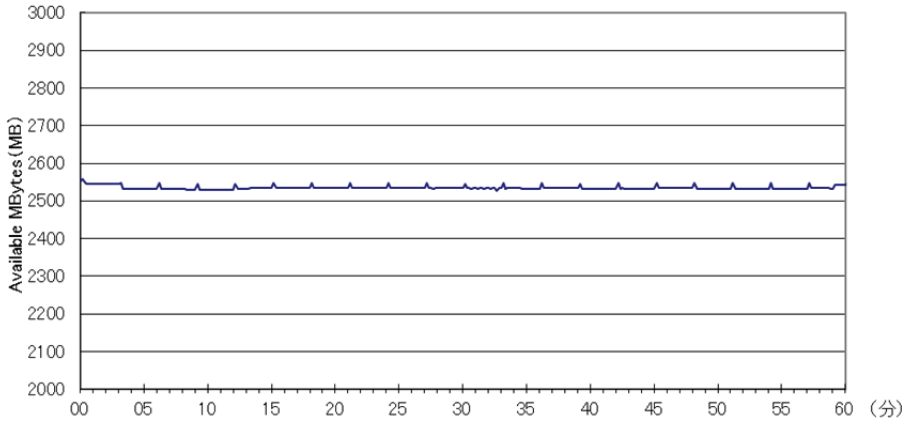


図-5.50　プロジェクト数 5

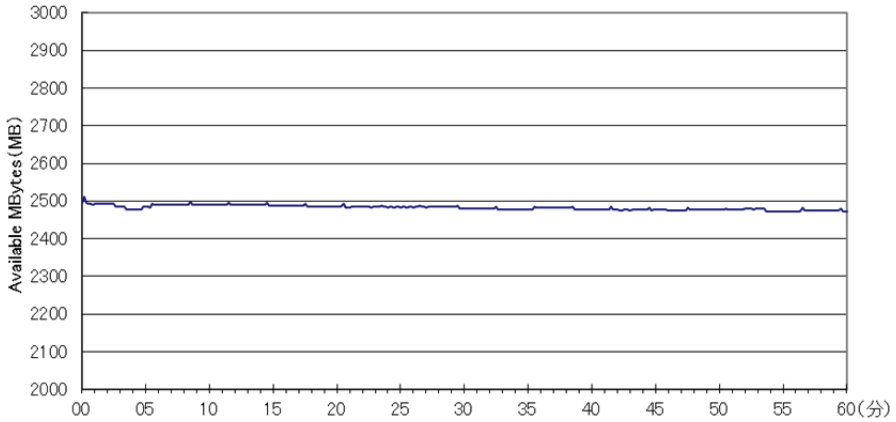


図-5.51　プロジェクト数 10

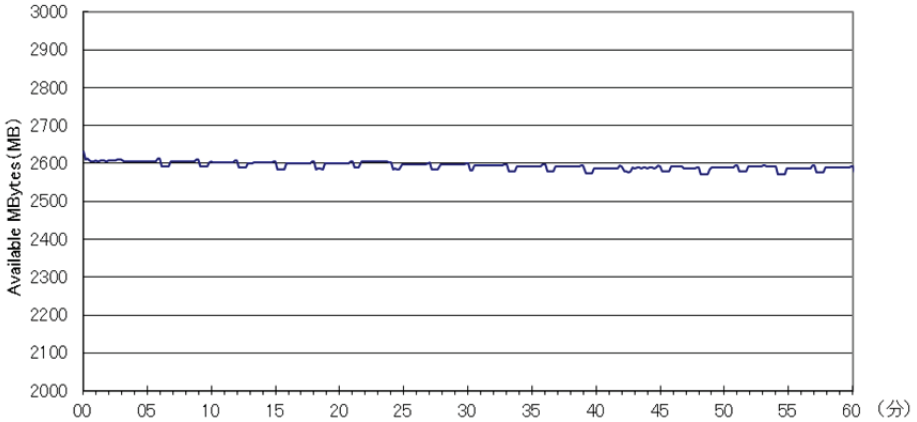


図-5.52　プロジェクト数 15

利用可能メモリ（Available Memory）の負荷試験結果（図-5.50 から図-5.52）

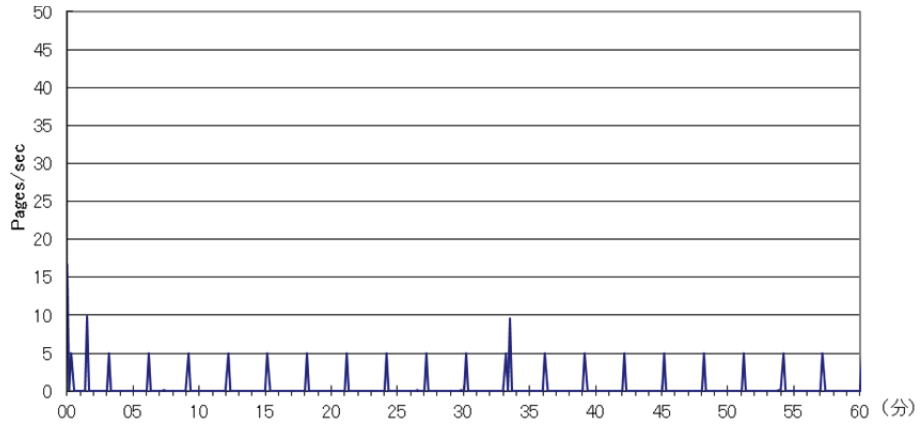


図-5.53　プロジェクト数 5

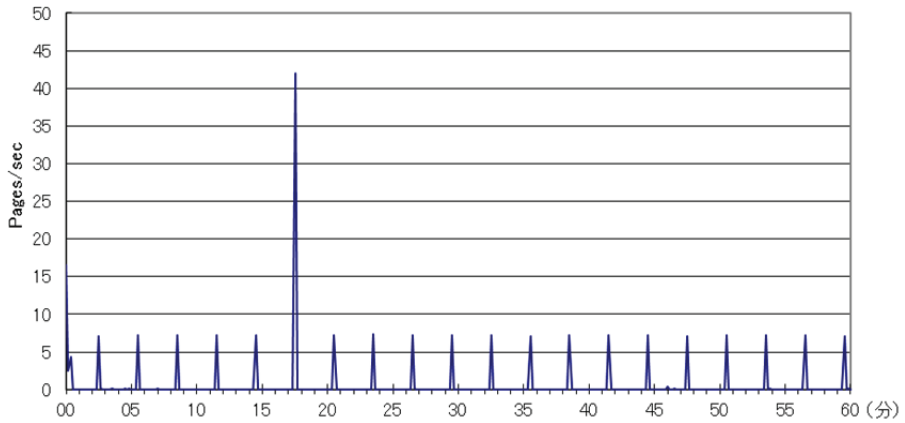


図-5.54　プロジェクト数 10

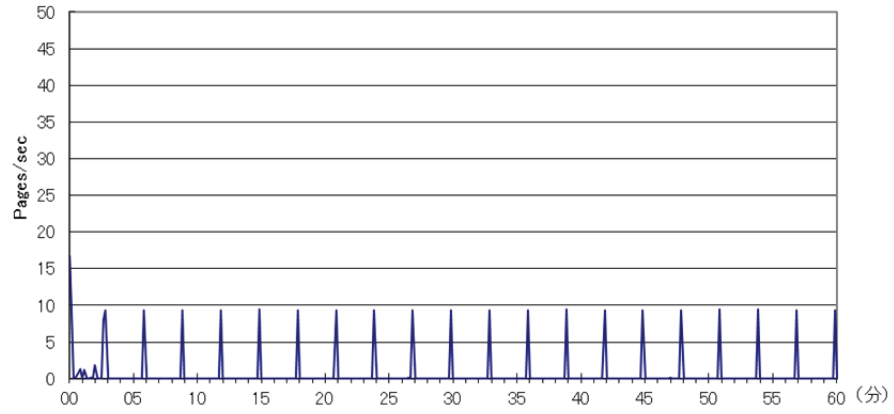


図-5.55　プロジェクト数 15 の Pages/sec

Pages/sec（ハードページフォルトを解決するための HDD とのデータ授受）の負荷試験結果（図-5.53 から図-5.55）

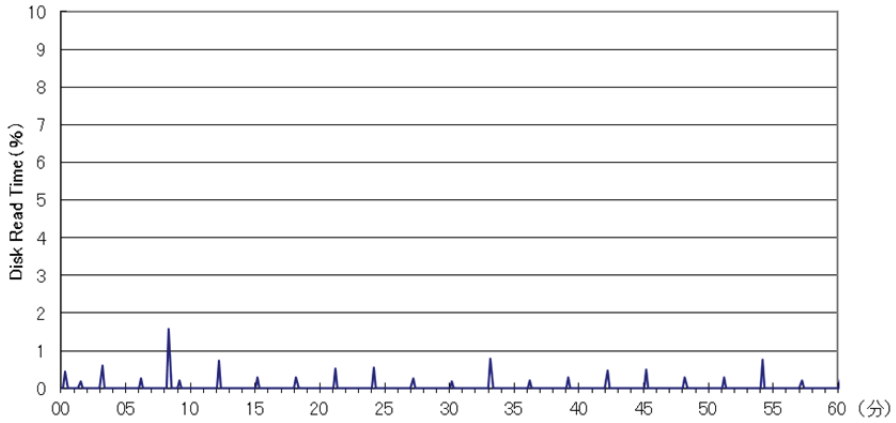


図-5.56　プロジェクト数 5

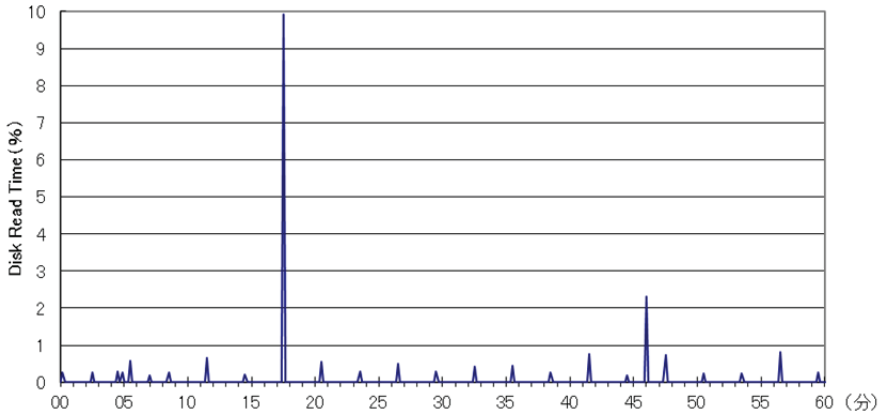


図-5.57　プロジェクト数 10

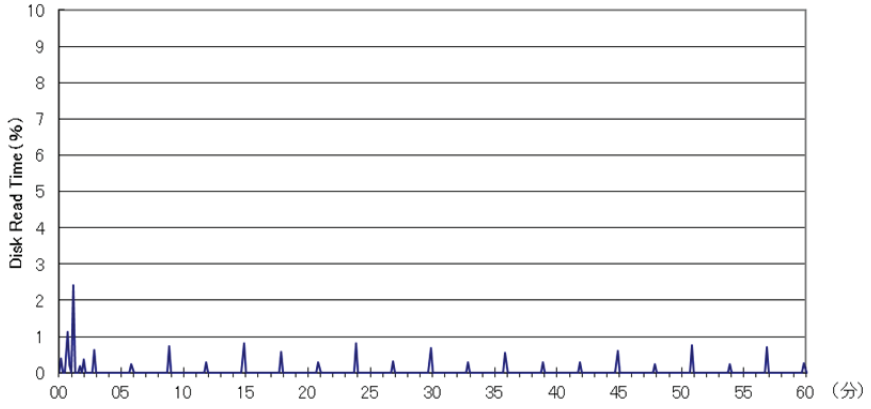


図-5.58　プロジェクト数 15

Disk Read Time（読み込み要求でディスクがビジー状態だった割合）の負荷試験結果
（図-5.56 から図-5.58）

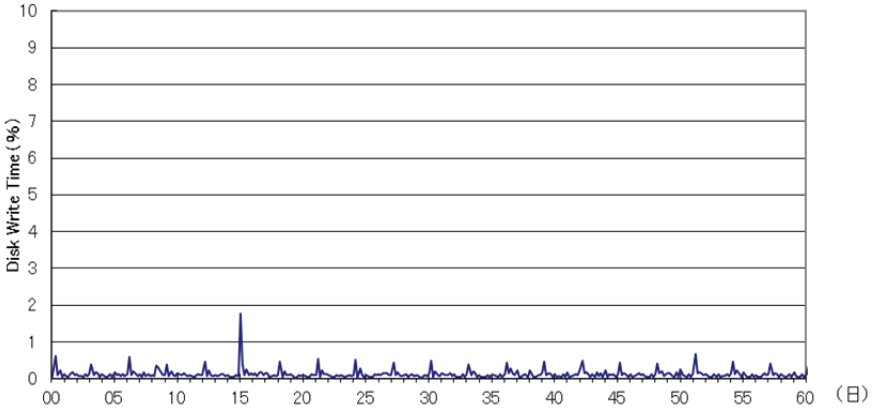


図-5.59　プロジェクト数 5

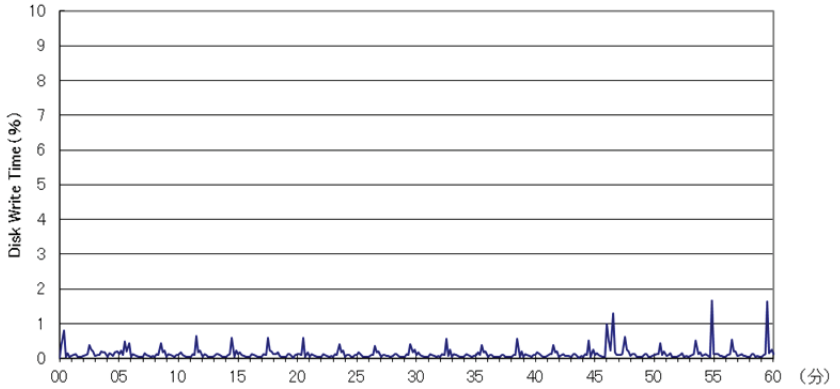


図-5.60　プロジェクト数 10

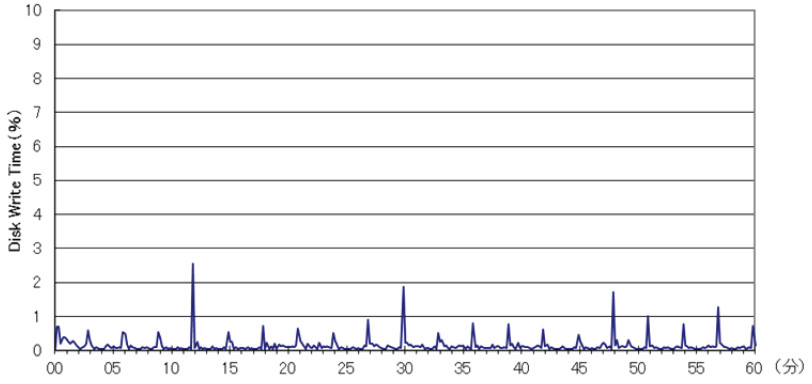


図-5.61　プロジェクト数 15

Disk Write Time（書き込み要求でディスクがビジー状態だった割合）の負荷試験結果
（図-5.59 から図-5.61）

第 6 章　結論

6-1 概説

本研究では，日本国内の水理・水文解析をとりまく課題を分析した上で，既存の国内外の汎用プラットフォームの開発の経緯や開発仕様を調査し，それらの汎用プラットフォームの国内の課題に対する適応性を分析した．これらの分析結果を踏まえて，国内の課題を解決するために新たに開発する汎用プラットフォームの開発方針および機能要件を分析し，機能要件を実現するためのプログラム設計を行い，実装した．また，汎用プラットフォームの公表後の普及状況等の分析を行い，今後の汎用プラットフォームの開発・運営プロジェクトの今後の展開について考察した．

6-2 結論

本研究によって，下記に挙げる知見を得た．

① 河川・流域に関するデータについては，整備が進む方法に向かっている．データを提供するための標準インターフェースについても一部実装されている．しかしながら，データの整備については，河川事務所により行われているところであるが，作業状況は，厳しくなりつつあると推察される．

② 国内における水理・水文解析ソフトウェアについては，OHyMoS 以外に組織を越えた連携の動き認められなかった．データとの連携性も悪く，データの整備に解析ソフトが追いついていない状況である．

③ 河川技術者の状況については，人員削減の中，水理・水文解析に労力をかけるのが厳しい状況になってきていると推察される．また，組織間をつなぐ共通のツールがないので，研究機関の知見を事業の現場に活用しにくい状況があると推察される．

④ 既存の汎用プラットフォームとして，OMS，OpenMI，OHyMoS を調査した．OpenMI については商用ベースのソフトウェアを対象としていること，OMS および OHyMoS については研究者ベースのプラットフォームであり，多様なユーザを対象とした汎用プラットフォームはなかった．

⑤ 日本国内の水理・水文解析をとりまく状況および既存の水理・水文解析プラットフォームの開発状況を勘案して，本プロジェクトで開発する水理・水文解析のための汎用プラットフォームの開発目的を「水理・水文解析モデルの研究開発の促進」，「河川事業等への研究成果の反映の迅速化」および「河川技術者等の技術力向上」とし，開発方針を①解析モデル開発の省力化・効率化，②モデル開発者の権利保護，③計算の透明性の確保，④多様なユーザへの対応とした．これらに対応するための機能要件を分析した．

⑥ 分析結果から得られた機能要件を実現するためのプログラム設計を行った．演算制御方式を２種類および要素モデルのタイプを２種類定めるとともに，要素モデル間収束計算

の方法を定めた.また,要素モデルのクラス設計および演算処理の方法の設計を行った.

⑦ 要素モデルのラッピング方法としての機能要件を検討するとともに,代表的なラッピング手法として名前付きパイプ方式と動的リンク方式を取り上げ，Fortran ラッパーの設計・実装および性能試験を行った.その結果,名前付きパイプ方式は簡易な方法であり，オリジナル Fortran プログラムの修正は少なくて済むが，要素モデルの接続や演算時間間隔設定の自由度が低く，演算速度も遅いことが確認された．動的リンク方式は，オリジナル Fortran プログラムの修正量は多いが，接続や演算時間間隔設定の自由度は通常の要素モデルと変わらないこと，演算速度が速いことが確認できた，また，内部状態量を持つ Fortran 演算プログラムに対しては正しい演算結果を得るためには，ラッピング要素モデル側で状態量を保存する必要があることが確認できた.

⑧ 機能拡張ツールの一つとして水文水質データ取得ツールの設計を設計・実装を行った.

⑨ マルチプロジェクト演算の方法として,マルチスレッド方式とマルチプロセス方式をとり上げ，長所・短所を分析した．汎用プラットフォームのマルチプロジェクト演算方式としてマルチスレッド方式を選択し，マルチプロジェクト演算機能を設計・実装した．マルチプロジェクト演算機能の機能確認のために実流域を対象とした卓上簡易洪水予測システムを作成し，3 日間の試験運転を行い，計算機メモリや CPU に異常が発生しなかったことを確認した．また，負荷試験を行い同時並行で稼働するプロジェクト数を増やしても，PC にかかる負荷が著しく高くなるようなこともないことも確認した.

6-3 普及状況の分析

ここでは，新たに開発した汎用プラットフォームの公表後のダウンロード状況やユーザ属性に関する調査に基づき，その普及状況を分析する.

6-3-1 ダウンロード件数の推移

初版の CommonMP は，2010 年 3 月 31 日にリリースされ，CommonMP のウェブサイトからダウンロードできるようになっている．CommonMP をダウンロードする際は，CommonMP のウェブサイトにてユーザ登録しなければならない仕組みになっている．また，一部を除いて CommonMP のウェブサイト以外では CommonMP を配布していないので，ダウンロード件数の推移を見れば，一般に出回っている CommonMP の普及状況を把握することができる．下記に CommonMP のダウンロード件数の推移を示す（図-6.1）.

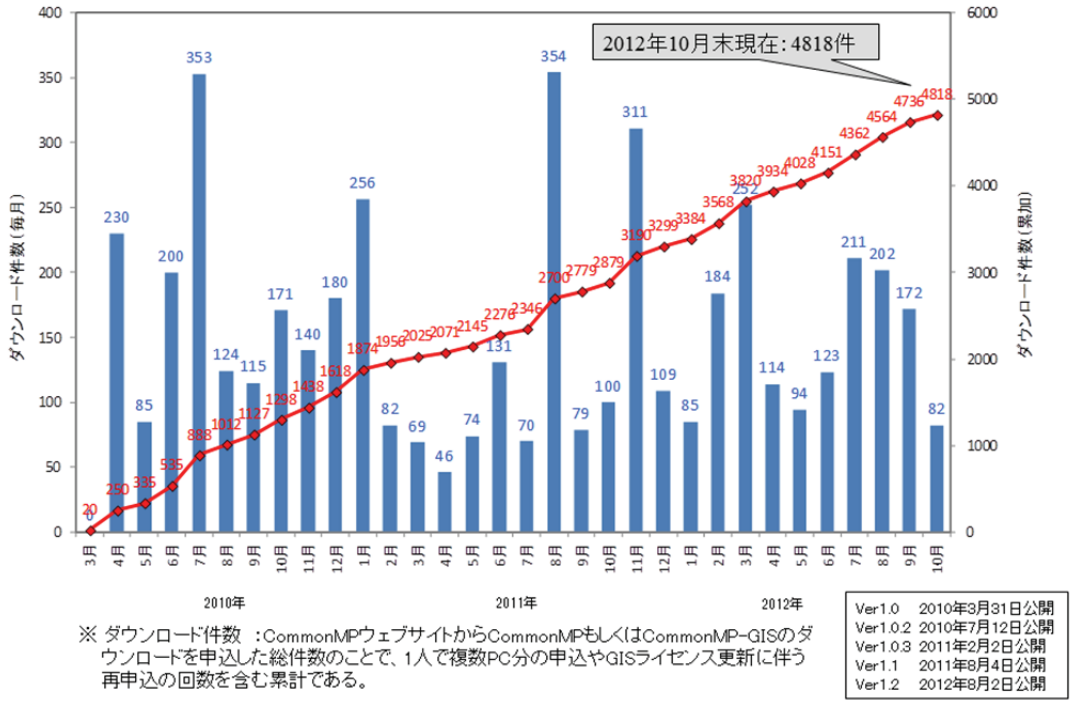


図-6.1　CommonMP のダウンロード件数の推移

ダウンロード件数には，CommonMP-GIS の 6 ヶ月ごとのライセンス更新分が含まれているので，純粋なユーザ数の増加はその分を差し引かなければならないが，ダウンロード件数は順調に伸びており，CommonMP のユーザ数もそれに応じて伸びていると考えられる．2012 年 10 月末時点でダウンロード件数は約 4,800 件に対して，ユーザ数は約 2,000 名である（電子メールアドレスベースでカウント）．月ごとのダウンロード件数は一定しておらず，新たなバージョンの CommonMP の公表等のイベントがあった月のダウンロード件数が多くなっている．

6-3-2　ユーザ属性等の分析

CommonMP プラットフォームのダウンロード申請に際しては，ユーザの所属組織等の属性情報の登録や利用目的を選択するようになっている．所属属性の内訳は下記のとおりである（図-6.2）．おおよそ国の機関が 40%，民間が 40%，大学関係が 5%程度であり，この割合は初版のプラットフォームの公表以来ほとんど変わっていない．国の機関や民間に比べて大学関係のユーザのダウンロードが少ないと言える．

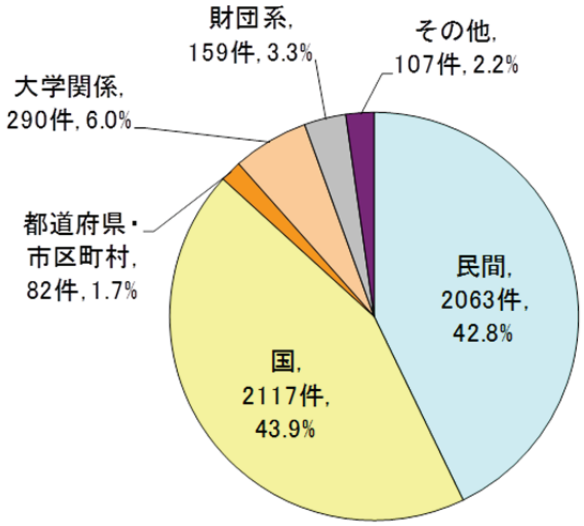


図-6.2 ダウンロードユーザの所属の内訳

2012 年 6 月からはプラットフォームのダウンロード申請フォームの修正し，それによりプラットフォームの利用目的や利用の動機をユーザに尋ねるようにしている．ダウンロード時の利用目的の内訳を下記に示す（図-6.3）．利用目的のうち，業務や研修・講習会に比べて，研究・学習が少ないことがわかる．自発的な利用というよりは，業務や業務遂行上必要なスキルの獲得のために利用していると分析できる．また，GIS ライセンス更新を除くその他の項目のうちでは，試用・テストが最も多く，特定の目的のために利用するユーザは少ないと推測できる．

【2012/6/01～2012/9/30】

利用目的(項目)	件数
業務	318
研究･学習	154
研修･講習会	230
要素モデル開発	25
システム開発	21
試用･テスト	75
データ取得	36
GISライセンス更新	49
その他	11

※利用目的(項目)は、複数回答あり

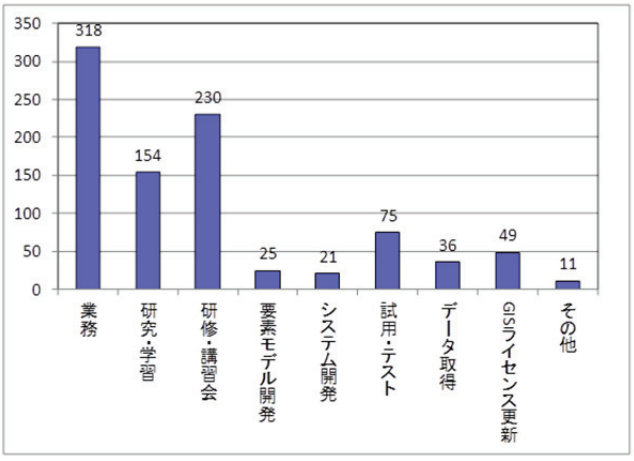


図-6.3　CommonMP の利用目的（複数回答）

次に CommonMP 利用の動機を集計した結果を下記に示す(図-6.4).利用の動機については，

「要素モデル開発」,「CommonMP の機能をもっと活用したい」および「データベースからのデータ取得」が目立って多い．前２者については，CommonMP の機能活用や要素モデル開発のためのスキルの学習のであり，実務への活用ではなくて，依然学習段階のユーザが多いことを示していると解釈できる．「データベースからのデータ取得」については，水文水質データ取得ツールの活用のことを指しているものであり，観測データの取得に関するニーズが高いことを示していると言える．

【2012/6/01～2012/9/30】

CommonMP利用の動機	件数	割合(%)
CommonMPで自分の要素モデルを開発したい	199	28.1
自分のモデルを他の人に使ってもらいたい	8	1.1
自分のモデルを他の人のモデルと組み合わせて使いたい	8	1.1
データベースからデータ取得ツールでデータをダウンロードしたい	93	13.1
使っているFortranプログラムをCommonMP上で使いたい(ラッピング)	14	2.0
CommonMPの機能(GUI、CUI)をもっと活用したい	137	19.4
その他	249	35.2
合　計	708	1000

※利用の動機は重複なし

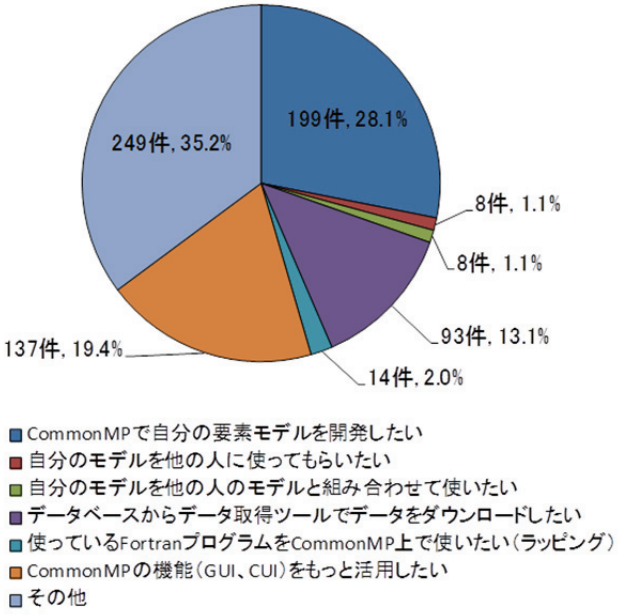


図-6.4　CommonMP 利用の動機

6-4 今後の展開

ここでは，前章までの調査・検討・分析を踏まえて，今後の汎用プラットフォームの開発・運営プロジェクトの展開について考察した．

6-4-1　仕事のやり方を変える

CommonMP の開発目標の一つである「河川事業等への研究成果の迅速化」を実現するため，事業の現場（河川事務所等），研究の現場（大学等），解析の実行部隊（民間コンサルタント）の間で連携がよくなるよう，現状の河川事務所が実施している解析業務のやり方を変える必要がある（図-6.5）．現行では，河川事務所等が河道計画検討を発注する場合，だいたい民間コンサルタントが受注して実施している．一旦，業務を受注した民間コンサルタントは，独自の解析ソフトウェアを使い，解析を行い，解析結果を事務所に納めて業務を完了する．河川事務所は，解析モデルを持っていないので，業務完了後は再計算もパラメ

ータ等を変えた感度分析も実施することができない.

一方，CommonMP を導入した後は，河川事務所の技術職員も受注する民間コンサルタントも CommonMP を利用することとなる．発注側と受注側が同じ解析ソフトウェアを使うので，解析中のシミュレーション・プロジェクトをやり取りしながら，業務を進めることができる．業務完了後も再度シミュレーションをして，解析結果を見直すことも，パラメータを変更して,感度分析を実施することもできる.再度同様の業務を発注する場合においても，事前に自前で概略検討することができる．また，河川事務所等の技術職員が自ら解析する機会が増えるので，技術力向上が期待できるし，水文データや河道断面データに触れる機会が増え，自ら解析を行うことから，高い動機付けを得ることとなり，データの誤りを発見・修正するなどのデータの品質向上にもつながる.

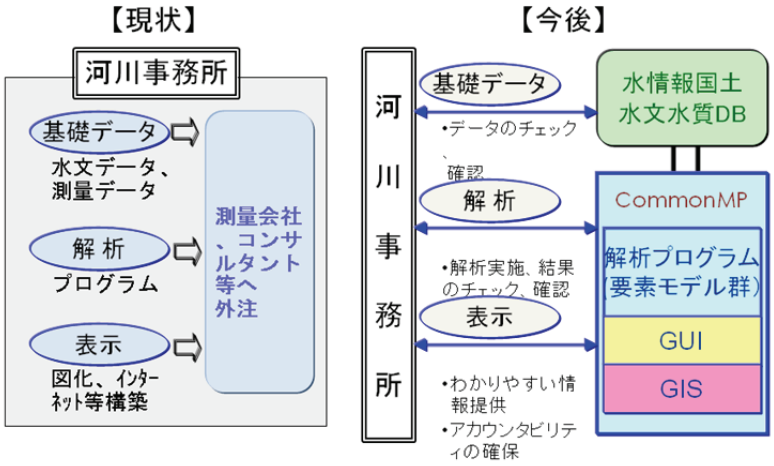


図-6.5　仕事のやり方を変える

6-4-2　洪水予測システム（基幹システム）への導入

現場河川事務所におけるスタンドアロン運用の CommonMP を用いた解析業務の他に,洪水予測システム等の基幹システムで用いることを目指す（図-6.6）．例えば，デスクトップにはすでに CommonMP は導入してあると仮定して,河川事務所等で行われている洪水予測において,CommonMP を導入することとする．基幹システムは,24 時間 365 日稼働（サーバ運用）が普通であるので，パラメータの感度分析を行い，精度を向上させるのは容易でないが，デスクトップの CommonMP で基幹システムと同じモデルを使うことにより,デスクトップの CommonMP で過去データやリアルタイムデータを用いてパラメータ調整した結果を基幹システムに反映することができる．これを予測実務に適用した結果は，学術的にも貴重な情報になる．大学等の研究機関においても同一の解析モデルを使うことができれば，大学等の有する知見とこれらの情報を融合して，解析モデルの精度向上にもつなげることができる．「6-4-1 仕事のやり方を変える」で述べた方策と合わせて，技術者の技術力向上，デー

タの品質向上，解析モデルの品質向上，技術者・研究者のモチベーション向上というスパイラルアップが期待できる．

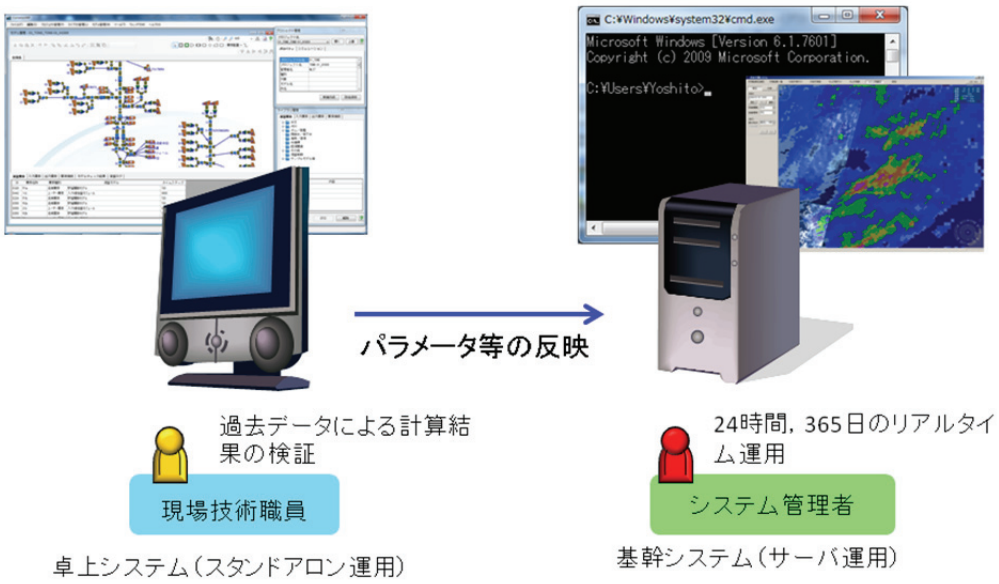


図-6.6　洪水予測システムへの導入

6-4-3　エディションの分化

CommonMP を「6-4-2 洪水予測システム（基幹システム）への導入」で述べた基幹システムとして導入するためには，CommmonMP のシステムソフトウェアとしての安定性・堅牢性を高める必要がある．また，洪水予測を行うためには，リアルタイム演算を行う必要があり，そのためには第 5 章で述べてマルチプロジェクト演算機能が必要である．現行の CommonMP Ver1.2 のマルチプロジェクト演算機能は，マルチスレッド方式で実現されており，スタンドアロン運用を行うデスクトップ・システムとしては適切な方式であるが，高い安定性・堅牢性が求められる基幹システムとしての運用には適してしない．より安定性・堅牢性が高いマルチプロセス方式で実現する必要がある．また，ソフトウェアとしての安定性・堅牢性を損なう要因となる GUI はサーバ運用では必要ないので削除すべきであり，代わりに他のソフトウェアと連携するための API が必要となる．このようにスタンドアロン運用する CommonMP とサーバ運用する CommonMP では，必要となる機能が大きく異なるので，それぞれデスクトップ・エディションとサーバ・エディションとして別のエディションとして管理するのが適切である（図-6.7）．デスクトップ・エディションやサーバ・エディションの他にも，特殊の用途に対しては，特定の機能を付加・削除したそれぞれの目的にあったエディションに分化されるのが，ユーザの利用上は望ましい．

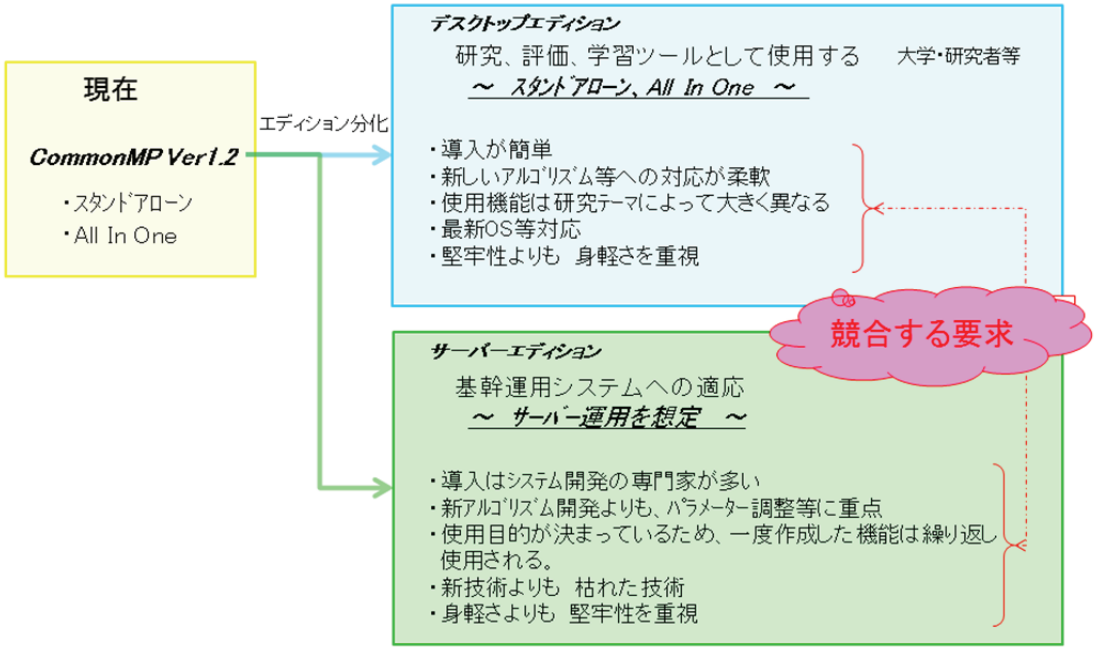


図-6.7　デスクトップ・エディションとサーバ・エディションの分化

6-4-4　海外展開

アジアモンスーン地帯において，我が国の水理・水文ソフトウェアは比較優位があると考えられ，CommonMP 開発・運営プロジェクトにおいては，当初から海外展開を目標としていた．日本製のソフトウェアは，個々の解析モデルについては，必ずしも外国製の解析モデルに劣るものではないが，他のモデルと連結できるようなプラットフォーム化に対応していなかったため，海外製のソフトウェアに活用実績が少ないものとなっていた．また，国内においては，民間コンサルタントの場合，互いに競争相手となるため，協力体制を築きにくい状況であるが，CommonMP の海外展開は，国内企業が協力し合うためにもよい機会となる．

6-4-5　自立発展性の確保

現在のところ，CommonMP の開発およびプロジェクト運営経費は，ほとんどすべて国総研からの予算で賄われている．ソフトウェア開発の場合，それ自体の機能を改良しなくても，OS 等の周辺技術の進歩に合わせて改良して必要があり，また，ソフトウェア維持管理のために専門的な知識を者を確保するための経費は一定程度は確保していく必要がある．ソフトウェアの維持管理やプロジェクトの発展的な継続のためには，国総研がもつ予算以外のスキームの活用や，国総研以外の者でも，プラットフォーム本体の開発が進められるようなライセンス形態の検討，要素モデルの有償化，書籍の販売，現在は無償で提供している

研修・コンサルティング等から収入を得ること，収入の受け皿となる組織づくり等が必要となる．また，要素モデルの有償化やCommonMPとのバンドル版の再配布等に当たっては，ますます要素モデルやプラットフォームの著作権等の整理が重要になり，要素モデルやプラットフォームのライセンスの設定を厳密に行なっていく必要がある．

参考文献

1) 藤田光一，小路剛志，吉谷純一：水理・水文・水質シミュレーションモデル・ソフトウェアの開発戦略に関する調査報告書，国土技術政策総合研究所資料 No.410，国土技術政策総合研究所，2007.
2) 高棹琢馬，椎葉充晴，市川温：構造的モデリングシステムを用いた流出シミュレーション，水工学論文集，第 39 巻，(社) 土木学会，pp.141-156，1995.
3) 湧川勝己,佐古俊介:河川計画シミュレーターの概要について,JICE REPORT vol11/07.03,（財）国土技術研究センター，pp.61-62，2007.
4) 菊森佳幹：水・物質循環解析のための汎用プラットフォームの開発，河川 No.762,(社) 日本河川協会，pp.31-35，2010.
5) 国土技術政策総合研究所:CommonMP ウェブサイト,URL:http://framework.nilim.go.jp
6) 菊森佳幹：水・物質循環解析のための汎用プラットフォームの開発運営に関する協定の締結，土木技術資料，(財) 土木研究センター，pp.44，2010.
7) 菊森佳幹，川戸渉，吉谷純一：水理・水文解析のための汎用プラットフォームの開発方針および機能要件の分析，土木学会論文集 B1（水工学）Vol.69 No.1 pp.1～13，土木学会，2013.
8) 国土交通省：水文水質データベース，URL:http://www1.river.go.jp
9) 水文観測業務規程細則第 7 条第 2 項，第 22 条，第 14 条第 2 項，国土交通省，2002.
10) 国土交通省：川の防災情報，URL: http://www.river.go.jp
11) 国土交通省：水情報国土データ管理センター，http://www5.river.go.jp
12) 竹本典道，小川鶴蔵，佐藤宏明，本間君枝：水情報国土の全体構想と活用，河川情報シンポジウム講演集，(財) 河川情報センター，pp.5-1～5-9，2009.
13) 河川 GIS・河川アプリケーション標準インターフェースガイドライン，国土交通省河川局，2006.
14) 河道計画検討の手引き（財）国土技術研究センター編，pp.110，山海堂，2002.
15) 平成 23 年度版公務員白書，人事院，pp.209，2011.
16) 菅昌徹治：EU の水管理政策【第 1 回水枠組指令と洪水指令】，河川 2012-1 月号，pp.86-93，(社) 日本河川協会，2012.
17) P.J.A. Gijsbers, R.V. Moore, C.I. Tindall: HarmonIT: Towards OMI, an Open Modelling Interface and Environment to harmonize European developments in water related simulation software, HydroInformatics, 2002.
18) United States Department of Agriculture, Collaborative Software Development Laboratory, URL:https://colab.sc.egov.usda.gov/
19) 立川康人，佐山敬洋，宝馨，松浦秀起，山崎友也，山路昭彦，道広有理：広域分布型

物理水文モデルを用いた実時間流出予測システムの開発と淀川流域への適用，自然災害科学 26-2，pp. 189-201，2007.
20）ラッピングマニュアル，国土技術政策総合研究所，pp. 3-5，2010.
21）菊森佳幹，飯田進史，荒木千博，米勢嘉智，川戸渉：リアルタイム洪水予測への対応のための CommonMP の機能改良，土木学会第 67 回年次学術講演会概要集，（社）土木学会，2012.
22）北川明：統一河川情報システムの運用と展望，河川情報シンポジウム講演集，（財）河川情報センター，2009.

謝辞

本研究（プロジェクト）は，日本発の水理・水文解析のための汎用プラットフォーム（CommonMP）を開発し，普及させようという極めてチャレンジングな試みです．筆者がこのプロジェクト関わり始めたのは，このプロジェクト予算化されて本格的に始動した 2007 年（平成 19 年）4 月からでした．それから 6 年が経ちました．その間に色々なことがありました．2007 年 9 月には，このプロジェクトのキックオフ・ミーティングというべき土木学会全国大会の研究討論会を広島大学で開きました（以降，研究討論会は毎年開かれています.）．研究討論会の開催に間に合わせるために，CommonMP のウェブサイトを急ごしらえで開設しました．2010 年 3 月には，コンピュータサイエンス専攻で修士号をとりました．畑が違うので苦労しました．修士課程での最初の講義で，プログラマー35 歳定年説[注1)]の説明を受けて，面食らいました（35 歳をとうに過ぎている筆者はどうしたらいいのでしょうか？）． 2 回り近く年齢が離れている学生と机をならべて勉強したり，Emacs エディタ[注2)]のわかりにいくキーバインドを必死になって覚えたりしたのを，今でも昨日のことのように思い出します．同じく，2010 年 3 月には，CommonMP Ver1.0 を公表し，ソフトウェアの管理が始まりました．今は，ウェブサイトの管理とソフトウェアの管理，要素モデルの管理，そして研修講師，新たなスキルの獲得と，とてもすべてには手が回らず，四苦八苦しています．

このプロジェクトが予算化されて本格始動する前から汎用プラットフォームの構想自体は動いており，大昔から関わっているメンバーがいます．中央大学の山田正教授には汎用プラットフォームの構想を強力に推進していただきました．山田先生がいなかったら CommonMP プロジェクトの予算化・本格始動はなかったかもしれません．京都大学の椎葉充晴教授は，OHyMoS の生みの親であり，CommonMP の開発や本論文執筆に当たり多大なる指導をいただきました（OHyMoS は，CommonMP の兄貴分に当たる関係になります.）．椎葉先生がいたので，CommonMP が国産の汎用プラットフォームになり得たものと思います．京都大学の立川康人准教授には，技術的な指導をいただくとともに，研究討論会等のイベントの企画や学会との調整に本プロジェクトのアクティブメンバーとしてよく動いていただきました．立川先生には，今後とも，本プロジェクトの一翼を担ってほしいと思います．河川情報センターの小川鶴蔵審議役（当時）には持ち前のフットワークの軽さで貴重な情報の提供や色々な人を紹介していただきました．退職されてからも推進委員会の委員として復帰していただいたときは，本当に嬉しかったです．この他にも，ソフトウェアの設計や実装をサポートしていただいた日立製作所や建設技術研究所の皆様，推進委員会の運営やウェブサイトの管理をサポートしていただいている河川情報センターの皆様，予算措置していただいている国土交通省水管理・国土保全局の皆様，プロジェクト推進委員会の皆様，その他このプロジェクトの関わってくださっている方々，本当に感謝しています．

今ここに博士論文を提出することができるのは，皆様のお陰です．深く感謝の意を表します．ありがとうございました．

菊森　佳幹（よしと）

注１）プログラマー35 歳定年説：プログラミング技術は進歩が激しく，技術の陳腐化も著しいため，常に新しい技術に目を向け習得していくバイタリティや，場合によっては永年の努力によって培ってきた技術を捨て去る柔軟性が必要である．また，年功序列的賃金体系のもとでは，高年齢のプログラマーはコストが高すぎると考える企業がある（特にプログラミングを単純作業と考える企業に多い）．俗に IT 土方とも呼ばれデスマーチとなった場合は徹夜が続いたり体力が必要となってくる。そのため、プログラマーとしての限界は 30～35 歳前後であるという説が存在した（ウィキペディアより抜粋）．

注２）Emacs エディタ：vi エディタとならび，Linux 上の代表的なテキストエディタの一つ．端末からの Linux サーバ等のテキストの編集には，どちらかのテキストエディタの習得が必須となる．マウスも効かず，メニューもないので，すべての操作をキーボードから行わなければならない．